(2)

Defense Nuclear Agency
Alexandria, VA 22310-3398

DNA-TR-89-25

HYPUF Upgrade Final Report

D. L. Johnson
McDonnell Douglas Corp.
5301 Bolsa Avenue
Huntington Beach, CA 92647-2048

February 1990

Technical Report

CONTRACT No. DNA 001-85-C-0363

Approved for public release;
distribution is unlimited.

Destroy this report when it is no longer needed. Do not return to sender.

PLEASE NOTIFY THE DEFENSE NUCLEAR AGENCY, ATTN: CSTI, 6801 TELEGRAPH ROAD, ALEXANDRIA, VA 22310-3398, IF YOUR ADDRESS IS INCORRECT, IF YOU WISH IT DELETED FROM THE DISTRIBUTION LIST, OR IF THE ADDRESSEE IS NO LONGER EMPLOYED BY YOUR ORGANIZATION.

DNA PRINTING PLANT

CUT HERE AND RETURN

DISTRIBUTION LIST UPDATE

This mailer is provided to enable DNA to maintain current distribution lists for reports. We would appreciate your providing the requested information.

□ Add the individual listed to your distribution list.

□ Delete the cited organization/individual.

□ Change of address.

NAME: ______________________________

ORGANIZATION: ______________________________

OLD ADDRESS	CURRENT ADDRESS
______________	______________
______________	______________
______________	______________

TELEPHONE NUMBER: (   ) ______________________________

SUBJECT AREA(s) OF INTEREST:

______________ ______________

______________ ______________

______________ ______________

DNA OR OTHER GOVERNMENT CONTRACT NUMBER: ______________

CERTIFICATION OF NEED-TO-KNOW BY GOVERNMENT SPONSOR (if other than DNA):

SPONSORING ORGANIZATION: ______________

CONTRACTING OFFICER OR REPRESENTATIVE: ______________

SIGNATURE: ______________

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY
N/A since Unclassified

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE
N/A since Unclassified

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
MDC-H5234

5. MONITORING ORGANIZATION REPORT NUMBER(S)
DNA-TR-89-25

6a. NAME OF PERFORMING ORGANIZATION
McDonnell Douglas Corp.

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
Defense Nuclear Agency

6c. ADDRESS *(City, State, and ZIP Code)*
5301 Bolsa Avenue
Huntington Beach, CA 92647-2048

7b. ADDRESS *(City, State, and ZIP Code)*
6801 Telegraph Road
Alexandria, VA 22310-3398

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*
SPWE/Wolf

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
DNA 001-85-C-0363

8c. ADDRESS *(City, State, and ZIP Code)*

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
63224C	SB	SC	DH009104

11. TITLE *(Include Security Classification)*
HYPUF Upgrade Final Report

12. PERSONAL AUTHOR(S)
Johnson, David L.

13a. TYPE OF REPORT
Technical

13b. TIME COVERED
FROM 860701 TO 881130

14. DATE OF REPORT *(Year, Month, Day)*
900201

15. PAGE COUNT
166

16. SUPPLEMENTARY NOTATION
This work was sponsored by the Defense Nuclear Agency under RDT&E RMSS Code B36407664D SB SC 00040 H2590D.

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
12	5	
20	11	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Stress Wave Response, Bade Geometric Dispersion,
Elastic-viscoplastic, Material Response Models,
Maxwell Dispersion, Ionization Equation of State. (jhd)

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

This report describes efforts accomplished to upgrade the public domain stress wave response code, HYPUF. The elastic-viscoplastic, Maxwell dispersion and Bade geometric dispersion material response models were implemented in HYPUF. Numerous errors in the equation of state and ionization equation of state routines were corrected. Finally, the code was restructured to facilitate implementation of a P-alpha material response model and incorporation of a tabular equation of state such as SESAME. It is recommended that a simple P-alpha model for the response of distended materials and the SESAME tabular equation of state be incorporated into HYPUF. Keywords,)

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Bennie F. Maddox

22b. TELEPHONE *(Include Area Code)*
(703) 325-7042

22c. OFFICE SYMBOL
DNA/CSTI

83 APR edition may be used until exhausted
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

PREFACE

This report describes work performed under SOW (Statement of Work) Task 8 of the AGT/UGT Correlation Contract, DNA001-85-C-0363. Under this task, modifications to the public domain coded HYPUF were performed. The need for the modifications was established under a related contract, Solid Booster/PBV Response.

Table of Contents

Section Page

SECTION 1
INTRODUCTION AND OVERVIEW

HYPUF is a stress wave response code that has the ability to calculate ionization effects in high temperature, high density plasmas. As such, HYPUF is a derivative of the PUFF-66 code. HYPUF is also a public domain code which means it is freely available to any defense contractor having a need to calculate the response of materials to radiation induced stress waves.

The modifications to HYPUF described in this report are part of a continuing program to provide a code suitable for analysis of material interaction with x-ray lasers and other high intensity radiation sources. Previous modifications included automatic zoning, rezoning and spall (fracture) capabilities. The modifications in this report include elastic-viscoplastic, Maxwell dispersion, and Bade geometric dispersion material response models, restructuring of the code to facilitate future modifications and numerous minor corrections to the equation of state and ionization equation of state subroutines.

The structure of the remainder of this report is as follows: Section 2 describes the implementation of the elastic-viscoplastic, Maxwell dispersion and Bade geometric dispersion material response models. Section 3 describes some corrections to the ionization equation of state and main equation of state subroutines. These corrections were necessary in order for HYPUF to correctly calculate bound-bound transitions in high Z materials such as gold and to treat hydrogen bearing materials. Section 4 describes the restructuring of the code. This restructuring was done to facilitate implementation of a P-α material response model for distended solids. Section 5 provides recommendations for future modifications to HYPUF.

SECTION 2
MATERIAL RESPONSE MODELS

This section describes the elastic-viscoplastic Maxwell dispersion and Bade geometric dispersion models which were implemented in HYPUF. All three models were incorporated as closely as possible to the way they were implemented in PUFF74. The only differences between the implementation in the two codes was that imposed by the fact that HYPUF is a temperature based rather than energy based code and that HYPUF has its equation of state package completely separate from the HYDRO routine.

The elastic-viscoplastic model is an extension of the elastic-plastic model which is used to calculate stress deviators in solid materials. In the elastic-viscoplastic model, the stress deviator can overshoot the yield surface value of $\frac{2}{3}Y$ (Y = yield strength). The stress deviator is computed incrementally from the differential equation.

$$\frac{\partial S_X}{\partial t} = \left\{ \begin{array}{l} \frac{4}{3}\frac{\mu}{v}\frac{\partial v}{\partial t}, \quad |S_x| \leq \frac{2}{3}Y \\ \frac{4}{3}\frac{\mu}{v}\frac{\partial v}{\partial t} - \frac{1}{T_r}\left(S_X + \frac{2}{3}Y\theta\right), \quad |S_X| > \frac{2}{3}Y \end{array} \right\} \tag{1}$$

In equation (1) above, the parameters are:

v = Specific volume

μ = Shear modulus which is calculated from its initial value μ_0 (input as AMU) based on the assumption that the ratio of shear modulus to bulk modulus is constant

Y = Yield strength which is calculated from the input parameters Y_0 (YO) and YADD as a function of plastic strain

T_r = Viscoplastic relaxation time calculated from the input parameters T_{ro} (TRELAX) and P_r (PRELAX).

θ = Dimensionless parameter of magnitude unity and opposite in sign to S_X.

In the above parameters, T_r is calculated from

$$T_r = T_{r0} \bullet \exp\left[-\frac{(|S_x|) - \frac{2}{3}Y)}{P_r}\right] \qquad (2)$$

The ratio of shear modulus to bulk modulus can be varied upon release, thus controlling the shape of the unloading path. This is controlled by the dimensionless parameter SHEARR. Other input parameters of interest are EM, the melt energy, and YMU the compression $\left(\frac{\rho}{\rho_o} - 1\right)$ at the elastic limit.

These parameters are used as described in Reference 1 and Reference 5.

The Maxwell dispersion model was originally designed to treat geometric dispersion in layered materials. In this model, a second stress deviator is calculated from the equation

$$\frac{\partial S_2}{\partial t} = \frac{4}{3}\,\frac{\mu_2}{v}\,\frac{\partial v}{\partial t} - \frac{S_2}{T_{r2}} \qquad (3)$$

Two inputs are required. They are:

μ_2 = Effective shear modulus, input as AMU2

T_{r2} = Relaxation time, input as TRELX2

The Maxwell dispersion model and the elastic-viscoplastic model may be used simultaneously. In that case, the total stress in the solid is given by

$$\sigma_x = P - S_x - S_2 \qquad (4)$$

Another geometric dispersion model available is the Bade geometric dispersion model. The Bade model was designed to describe the behavior of composites such as 3DQP, which have large scale heterogeneities. In this model, a rate dependent stress is calculated from

$$Q = \rho_o C_o^2 \bullet \left[\frac{2}{\omega_1^2} \frac{\partial^2 u}{\partial x \partial t} + \frac{2A}{\omega_1} \frac{\partial u}{\partial x} \right] \tag{5}$$

The required input parameters are ω_1 (input as OMEGA) and A (input as EQSTA). The Bade model can also be used in conjunction with the elastic visco-plastic model. The total stress is then

$$\sigma_x = P - S_x + Q \tag{6}$$

SECTION 3
CORRECTIONS TO EQUATION OF STATE

During the process of implementing the changes described in Section 2, some difficulties were encountered in using HYPUF. These difficulties included the inability to handle hydrogen bearing compounds, and the inability to handle TWCP or similar materials with a specific heat ratio in the vapor that is substantially different from $\frac{2}{3}$. In addition, Dr. Judy Gates of APTEK (Reference 2) discovered that HYPUF did not correctly treat the bound-bound transitions of high Z materials such as gold. This section describes the actions taken to correct these difficulties.

The errors discovered by Dr. Gates involved an inconsistency in the calculation of bound-bound transitions and the possibility of a floating point error in the calculation of the energy fraction for bound-bound transitions for high Z materials. In order to solve these problems, several changes were made. To correct the inconsistency in the number of bound-bound transitions which can be treated, the dimensions of NSAVE, EBB, and NLEC were increased as shown in table 1 below. In addition, the do-loop on index NTMP in subroutine OPAGUE (see Appendix A, page A-85) was changed to read:

```
     . . .
     DO 870 NTMP = 1, 15
     NTMP1 = 16 - NTMP
     IF (KI.LE.NLEC(N,NTMP1)) GO TO 890
870  CONTINUE
     . . .
```

Table 1. Change in dimensions of arrays used to calculate bound-bound transitions.

Common Block Name or Local	Array Name	Original Dimensions	Corrected Dimensions
AC	EBB	(10, 6, 14)	(10, 15, 14)
AC	NLEC	(10, 6)	(10, 15)
LOCAL*	NSAVE	(6)	(15)

*NSAVE is local to subroutine GENRAT.

In subroutine OPAGUE, RBB is the energy fraction for the bound-bound transition of interest. Originally, RBB was simply divided by the quantity (EION (N, JI) - ENIK (N, K1, K4)) in a loop to find the highest active edge contributing to the x-ray cross-section. However, in certain diabolical situations, there are no electrons being removed from subshell K4. In such a situation, (EION(N,JI) - ENIK (N, K1, K4)) which is the edge energy for removal of an electron from subshell K4, will be zero and a floating point error will result. To correct this problem, the integer array NSNIK (10, 8, 14) was added to common block AC and the following additions were made to the coding: An inner do loop was added after the inner loop to initialize NNIK (see Appendix A, page A-85):

```
      . . .
      DO 550 K=1, KI
      KL = NOEC(N) + 1 -K
550   NNIK (N,K1,KL) = NGRUP (K)
      DO 560 K = 1, 14
560   NSNIK (N,K1,K) = NSPDF (K)
      . . .
```

Then the following lines of code were added (see Appendix A, Page A-86).

```
         . . .
1010  RBB = EBB (N, NTMP, K4)
      KNN = KC + 1 - K4
      IF (NSNIK (N, K1, KNN) . EQ. 0) GO TO 1020
      RBB = RBB/(EION(N, JI) - ENIK (N, K1, K4))
1020  CONTINUE
         . . .
```

When the need to calculate the material response of hydrogen bearing materials was first encountered, the code would not cooperate. Numerous floating point errors made it necessary to alter the description of the elemental composition such that the presence of hydrogen was ignored. The primary source of these difficulties was traced to the indexing scheme used in the Saha equation solver and related sections of the code. The Saha equation solver is an indefinite loop designed to calculate the degree of ionization in each element in the material. Up to eight levels of ionization and the population fraction for each of the levels is computed. The Saha equation, however, is a relationship between two successive levels of ionization. Thus, the Saha equation must be used iteratively to find the eight most highiy populated levels of ionization and the fraction of the population in each level. According to Reference 6 the ratio of particle concentrations between the mth and m + 1st ionization levels is given by

$$\frac{\alpha_{m+1}\alpha_e}{\alpha_m} = \frac{1}{\rho N} K_{m+1}(T) \qquad m = 0, 1, 2 \ldots \tag{7}$$

where $\alpha_i = N_i/N$, and

$$K_{m+1}(T) = 2 \bullet \frac{U_{m+1}}{U_m} \bullet \left(\frac{2\pi m_e kT}{h^2}\right)^{3/2} e^{-\frac{I_{M+1}}{kT}} \tag{8}$$

$$\Sigma\alpha_m = 1 \tag{9}$$

$$\Sigma m \alpha_m = \alpha_e \tag{10}$$

In the above expressions, U_m and U_{m+1} are the electronic partition functions of the m and m+1 states, m_e is the mass of the electron, k = Boltzmann's constant, h = Plank's constant, T = temperature and I_{m+1} is the ionization potential of the m-ion (the increment in energy needed to remove the m+1st electron). For hydrogen, $\alpha_1 = \alpha_e = 1-\alpha_o$, and so equations (7) and (8) simplify to

$$\frac{\alpha^2}{1-\alpha} = 2\frac{U_1}{U_o} \bullet \frac{1}{\rho N}\left(\frac{2\pi m_e kT}{h^2}\right)^{3/2} e^{-\frac{1}{kT}} \tag{11}$$

In (11) α is the population fraction of the first ionization level. Thus, in order to enable HYPUF to handle hydrogen, two changes were required: First, the indexing scheme was altered so that the ground state or zeroth ionization level can be included in the calculations where appropriate. Second, the analytical solution to equation 11 was coded into the Saha equation solver, explicitly. Since the ground state was included in the eight possible levels of ionization subroutine FLOION also had to be modified to correctly calculate the ionization energy in the material. The changes in FLOION were (see Appendix A, page A-52)

```
      . . .
      DO 60 K=1, KN
      L = NI (K, N)
      IF (L.LE.0) GO TO 60
      EI (J) = EI (J)+R (K,N)*EN (N, L)*AF (M,N)*FLOAT (NATOM (M))
    1    *9.632E11 / XMW(M)
      IF (J.NE.JTS) GO TO 60
      IF (KPRIN.EQ.1) WRITE (6,190) MATL (M), NAMEL (N), KN,
    1    L, J, EI (J), R (K, N), EN (N, L), AF (M, N)
 60   CONTINUE
      . . .
```

In SAHA, the changes were (see Appendix A, page A-116, A-117):

```
      . . .
C
C     CHANGE IN KMAX TO INCLUDE GROUND STATE AS ONE OF THE
C     POSSIBLE IONIZATION STATES
C
      KMAX (N) = MIN0 (NTBL (N) + 1,8)
      KN = KMAX (N)
      . . .
      . . .
      90   DO 150 N1 = 1, NEM
              N = IELEM (M, N1)
              KN = KMAX (N)
              KGO = KN - 1
              IF (KN.LE. 2) GO TO 100
              GO TO 110
     100   CONTINUE
              R (1, N) - 0.
              A (1, N) = A (1, N)/ZSTAR*ZSTAR1
              IF (A(1, N) + GT. 0.) R (1, N) = 0.5 * A(1,N)*(SQRT (1.+ 4./A(1,N)) - 1.)
              IF (A(1, N) . GT. 1. .AND. R(1, N).EQ.0.) R(1, N) = 1.
              RMAX (N) = R(1, N)
              IF (KPRIN.EQ.1.AND. J.EQ.JTS) WRITE (6,240) NAMEL (N),
      1          ZSTAR, ZSTAR1, KN, KGO, A (1,N)
              GO TO 150
110   CONTINUE
      DO 120 K = 1, KGO
          . . .
```

Also, in SAHA, the logic to designate which ionization levels are of interest was changed (see Appendix A, page A-118)

```
. . .
DO 190 N1 = 1, NEM
N = IELEM (M, N1)
KN = KMAX (N)
IF (KN.EQ.1) GO TO 190

. . .
```

Once the changes described above were implemented and debugged, an attempt was made to calculate the response of TWCP. This attempt was initially met with frustration because of the way the equation of state routines handle expanded materials. In particular, if the ratio of specific heats of the vapor phase is substantially different from $\frac{5}{3}$, an error results. Under such circumstances, the code would attempt to converge to a negative temperature which was not allowed. Consequently, a negative density would eventually be calculated for a zone and a math error would result from the attempt to calculate the square root of a negative number. The correction was to change the coding in EQST (see Appendix A, page A- 42) and PE (see Appendix A, pages A-89 and A-90). In EQST the corrected coding is:

```
. . .
10  ALF = EQSTH (M) + (EQSTG (M) - EQSTH(M)) • SQRT (ENU)
    IF (ABS (ALF - EQSTH (M)) .LE. 1.E-3) GO TO 20
    GO TO 30
20  W1 = 1.
    GO TO 40
30  W1 = 2.* (EQSTG(M) - ALF)/(ALF - EQSTH (M))
    W1 = (2. + W1)/(1. + W1)
40  W2 = 1.5 * ALF * W1
. . .
```

The corrected coding in PE is similar to that in EQST:

```
. . .
50  ALF = EQSTH (M) + (EQSTG(M) - EQSTH (M)) * SQRT (ENU)
    IF (ABS (ALF - EQSTH (M)) . GT. 1.E-3) GO TO 60
    W1 = 1.
    GO TO 70
60  W1 = 2.* (EQSTG(M) - ALF)/(ALF-EQSTH(M))
    W1 = (2. + W1)/(1. + W1)
70  CONTINUE
. . .
```

With these changes HYPUF can now treat materials with any reasonable value for the ratio of specific heats.

SECTION 4
RESTRUCTURING

In order to simplify the task of implementing a P – α material response model or the SESAME tabular equation of state, it was decided to revise the structure of HYPUF. The structure of HYPUF at the beginning of this effort is shown in Figure 1. The final structure adopted is shown in Figure 2. With either structure, the subroutine GENRAT is called only once to set up the problem. If automatic zoning is requested, GENRAT calls subroutine AZONE, and AZONE in turn calls subroutine FINDRZ. GENRAT also calls subroutine FLOION to initialize the x-ray cross sections and determine the initial deposition profile. Since HYPUF does not have the option of using an arbitrarily specified deposition profile, FLOION is always called by GENRAT. The rest of the subroutines are called repeatedly as the program works through the solution to the equations. The names of the subroutines give a reasonable idea of their function. It is noteworthy that subroutine EQST calls subroutine FLOION to calculate the revised x-ray cross sections and determine radiation and thermal transport through the materials. In addition, subroutine REZONE calls subroutine EDIT only if the debug option is exercised.

In the restructured version of HYPUF, certain functions originally performed in subroutines EQST and FLOION were broken out into separate subroutines. Subroutines PT and PE were broken out of subroutine EQST in order to facilitate the implementation of a P-α model. Subroutines SAHA, OPAGUE, and TRANSP were broken out of subroutine FLOION in order to facilitate the implementation of the SESAME or other tabular equation of state.

Subroutines PT and PE are used by EQST to solve for the new pressure and temperature as a function of density and specific energy. Subroutine PT calls subroutine PE to determine pressure and energy as a function of density and temperature. Subroutine PT uses a Newton iteration with calls to PE using the latest guess at the correct temperature until the temperature is found that corresponds to the density and energy calculated by subroutine HYDRO. For each value of temperature and density supplied by PT, subroutine PE returns the corresponding specific energy and pressure. The final values of pressure and temperature are returned by subroutine PT to EQST.

Given the values of density and temperature, subroutine EQST calls subroutine FLOION to determine the transport of x-ray energy and heat. Subroutine FLOION now calls subroutine SAHA to determine the degree of ionization in the material. Subroutine SAHA uses the Saha equation iteratively to determine the degree of ionization. FLOION then calls subroutine OPAGUE to determine the revised electron structure in the atoms and the resulting corrections to x-ray cross section. FLOION then calls subroutine TRANSP to determine the electron thermal conductivity and Roseland mean opacity. This information is used to determine the transport of energy through the materials of interest and the revised deposition profile.

With the changes described above, it should now be relatively easy to implement a P-α model for distended materials and a tabular equation of state such as SESAME. For distended materials, the P-α model would be added to subroutine HYDRO. Subroutine HYDRO could then call subroutine PE iteratively to converge on a new value of α (the distension parameter). This avoids the need for a double iteration in the P-α (iteration on distension and iteration on temperature) which would otherwise be required.

A tabular equation of state model could be used in place of subroutines SAHA, OPAGUE and TRANSP if desired. Such a tabular equation of state could also be used to replace the functions of subroutines PE and PT. Since an appropriate tabular equation of state might not exist for all materials of interest, the use of the present structure of HYPUF should make it possible to treat combinations of materials where some materials would have a tabular equation of state and other materials need the analytical equation of state.

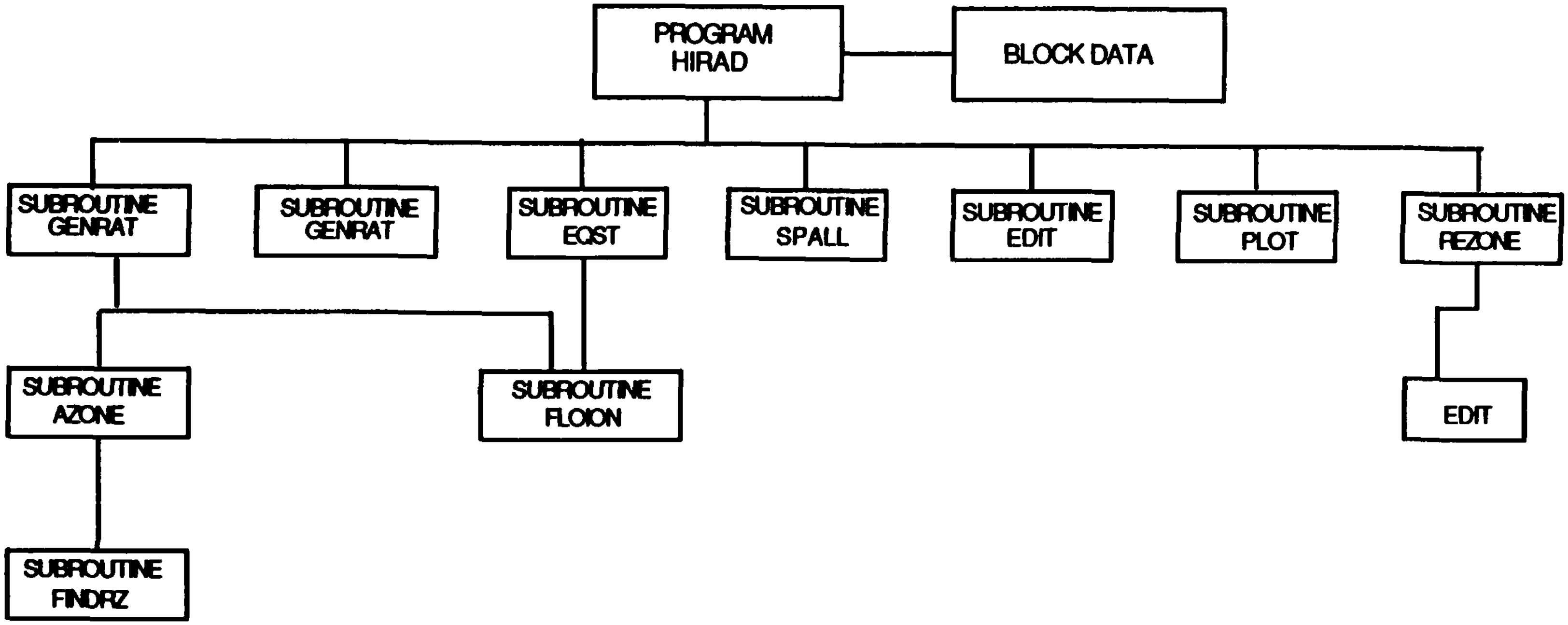


Figure 1. Structure of HYPUF at beginning of effort.

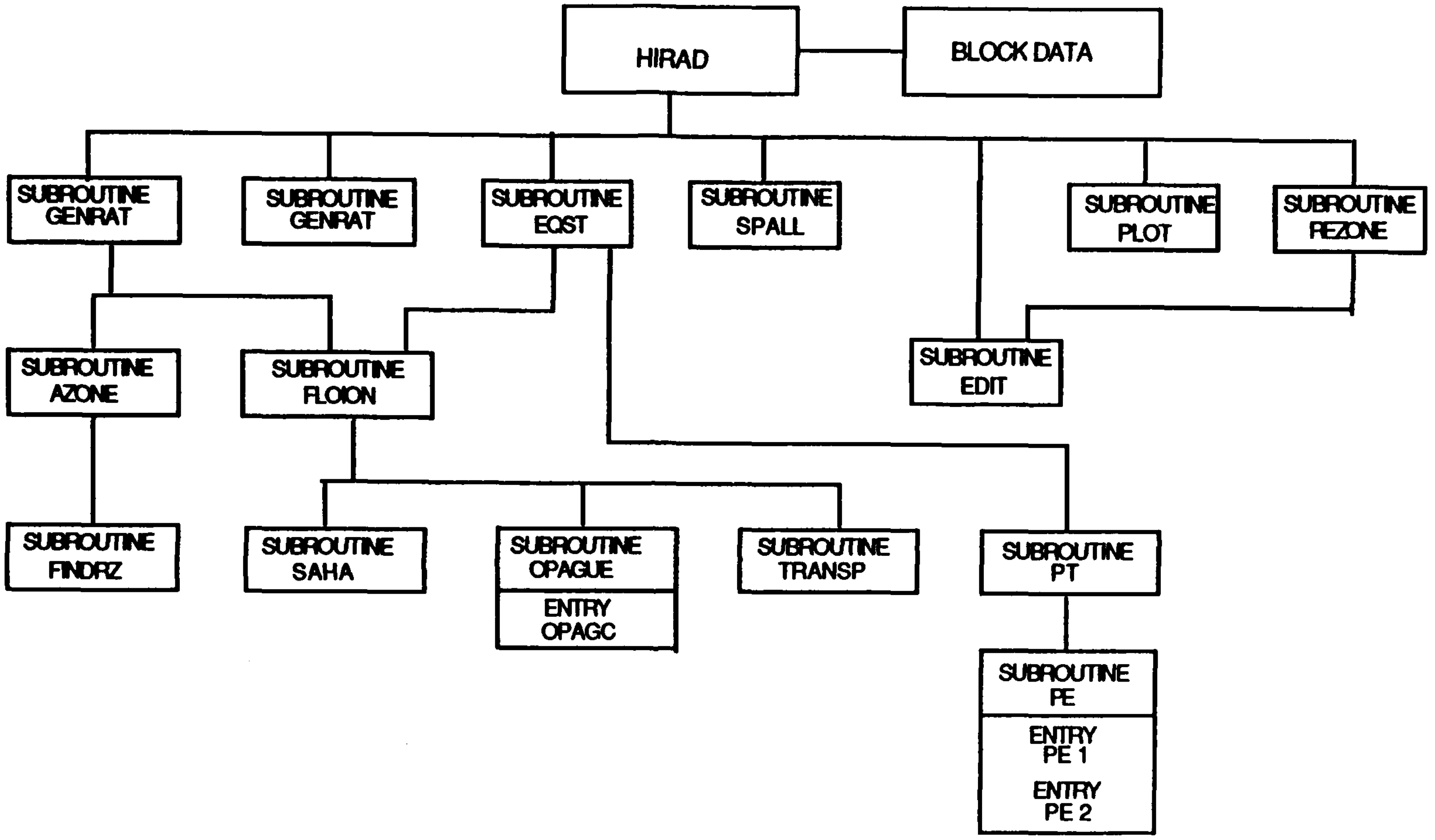


Figure 2. Revised structure of HYPUF.

SECTION 5
RECOMMENDATIONS

In order to make HYPUF useful for calculating the stress wave response of solid propellant booster materials, it is necessary to implement some type of P-α model for distended materials. Several such models are available, the simplest P-α known to the author is a tabular model developed by M. H. Rice of S-Cubed. The P-α model used in PUFF74 was developed by ETI (now GRC). In addition, there are some more elaborate models which have been developed by SRI International and implemented in SRIPUFF8. As a first step, it seems most reasonable to implement as simple a P-α model as possible in HYPUF. If subsequent experience demonstrates a need for a more elaborate model for distended materials, the ETI model in PUFF74 or one of the SRI models in SRIPUFF8 can be used.

There are situations in which the present analytical equation of state in HYPUF is not adequate. In particular, the inability of the equation of state to properly treat the behavior of liquids and the melting and vaporization phase transitions can produce significant errors in the calculation of the stress wave response of material where a large fraction of the material is melted. The simplest way to eliminate this difficulty is to implement a tabular equation of state such as the SESAME package developed by Los Alamos.

Recently, Dr. Judy Gates of APTEK (Reference 3), published a critique of the ionization equation of state in HYPUF. In her analysis, Dr. Gates recommended the use of the Plank mean opacities as well as the Rosseland mean opacities for calculating radiation transport. Such a modification would be simplified by the use of the SESAME equation of state since SESAME provides both the Plank and Rosseland mean opacities. Therefore, we strongly recommend the implementation of the SESAME equation of state in HYPUF.

SECTION 6
LIST OF REFERENCES

1. Robert Cecil, C., David Newlander, Capt., Richard J. Scammon, PUFF74 - A Material Response Computer Code, Volumes I and II (U), Air Force Weapons Laboratory (DYV). Kirtland AFB, NM. January 1977 (UNCLASSIFIED).

2. J. A. Gates, Private Communication (U), Aptek, Colorado Springs, CO. June 29, 1987 (UNCLASSIFIED).

3. J. A. Gates, Soft X-ray Absorbtion Coefficient Sensitivity Analysis (U), A-88-8R, Aptek, Colorado Springs, CO. June 1988 (UNCLASSIFIED).

4. P. J. Mallozzi, B. P. Fairand, and R. G. Jung, Effects of High-Intensity Photon Fluences (U), AFWL-TR-71-101, Air Force Weapons Laboratory, Kirtland AFB, NM. July 1972. (S-RD) (UNCLASSIFIED).

5. M. H. Rice, PUFF74 EOS Compilation (U), R-80-4296 (AFWL-TR-80-21), Systems, Science and Software, LaJolla, CA. August 1980 (UNCLASSIFIED).

6. Ya. B. Zel'dovich, and Yu. P. Raiser, Physics of Shock Waves and High-Temperatuce Hydrodynamic Phenomena, Volumes I and II (U) edited by Wallace D. Hayes and Ronald F. Probstein, Academic Press, New York, 1966 (UNCLASSIFIED).

APPENDIX A
HYPUF SOURCE LISTING

```
*COMDECK BLANK
C     BLANK COMMON
C
      COMMON CS(201), V(201), E(201), P(201), S(201), SD(201), U(201),
     1 ZM(201), TEMP(201), ZFM(201), X(201), Q(201), QO(201), YOZ(201),
     2 DV(201), EI(201), ITER(201), F(201), FO(201), EADD(201), ZF(10,20
     3 1), SS(201,3), TEMPO(201), TKEEP(201), ET(201), PN(201), XFX(201)
     4 , XFL(201)
C
      COMMON AMU(6), CUSP1(6), CUSPA(6), CUSPC(6), CUSPD(6), CUSPG(6),
     1 CUSPS(6), EQSTC(6), EQSTD(6), EQSTS(6), EQSTE(6), EQSTG(6), EQSTH
     2 (6), EQSTN(6), RHO(6), PMIN(6), LGDEL(6), YADD(6), YMU(6), YO(6),
     3 JBND(6), NELEM(6), XMW(6), IELEM(6,5), AF(6,10), XAW(10), NOE(10)
     4 , NTBL(10), XI(10,100), EN(10,100), JEDIT(10), JORG(10), TEDIT(25
     5 ), NZ(20), RZ(20), T(3), EE(3), START(3), SSTOP(3), NBB(3), NHNU(
     6 3), ES(3,109), AA(10,25), B(10,25), EDGE(10,25), NATOM(6), XCON(6
     7 )
C
      COMMON CKS, CO, C1, DTN, DTNH, IT, JCYCS, JFIN, JSMAX, JSMAXI,
     1 JRZL, JSTAR, JTS, JZPUL, LINE, LOZHIZ, N, NJEDIT, NMTRLS, NPRIN,
     2 NREZON, NRZ, NSPEC, NTAPE, NTEDT, PDTNEG, PDTPOS, SDURM, SK2M,
     3 SMAX, SSTOPM, TIME, TS, WTAPE, ILIN, ILOG, ICON, IDIF, ANGLE,
     4 DTMIN, DIFTST, DTPRIM, IFLOW, JHAT, NCOUNT, NDEP, JPRIN, ION,
     5 NDBG, TR
C
C     CHARACTER STATEMENTS
C
      CHARACTER *10 DISCPT, MATL, NAMEL
C
      COMMON /CHARB/ DISCPT(8), MATL(6), NAMEL(10)
C
*COMDECK AA
      COMMON /AA/ ITBL(96,6), ILTBL1(3), ILTBL2(3), ILTBL3(3), ILTBL4(1)
     1 , ILTBL5(3), ITABL(19), XNSTAR(14), TBL(109), SCALE(18), SCAX(15)
     2 , JTABL(14)
C
*COMDECK AB
      COMMON /AB/ ICHCK
C
*COMDECK AC
      COMMON /AC/ NOEC(10), EDGEC(10,20), EION(10,100), SCRENO(10,20),
     1 NGRUP(19), NSUM(19), NSPDF(14), NVARM(6), NVARE(10), NION(10,19),
     2 EBB(10,15,14), NLEC(10,15), NSNIK(10,8,14)
C
*COMDECK EQED
      COMMON /EQED/ RERAD
C
*COMDECK EQFL
      COMMON /EQFL/ ALF, ARG, ARGEXP, DEDT, DEDTP, DELRO, DES1, DFDT,
     1 DPDRC, DPDT, DPDTP, DTRC, DTRC1, DU, DXY1, DXY2, DXZ, EMU, ENU,
     2 ENU2, ES1, ES1N, ES10, EXPR, FXPC, IPLUS(10), IPLUSO(10), ITRY,
     3 KGO, KMAX(10), KN, NEM, NI(8,10), R(8,10), SK2M1, TEMPJ, TREF, V1
     4 , W1, W2, XITMP, XLAM1(201), XLAM2(201), XMAX, XMAX1, XNATOM, XN1
     5 , XP, XTEV, XX, XXX, XY1, XY2, XZ, XZ2, ZF1(10), ZSTAR, ZSTAR1,
     6 Z1, Z2
C
*COMDECK EQVP
      COMMON /EQVP/ AMU2(6), CH(6), EQSTA(6), GOKE(6), GOKE2(6), GOVERK(
     1 201), JB(6), MFLAG(6), OMEGA(6), PRELAX(6), QU(201), SD2(201),
     2 SHEARR(6), TRELAX(6), TRELX2(6), VAMU(201), YY(201), ZZ(201)
C
*COMDECK HYEQ
      COMMON /HYEQ/ JTRY
C
*COMDECK INDX
      COMMON /INDX/ I, ICOUNT, IGO, J, JCOUNT, J1, J2, K, KCOUNT, KPRIN,
     1 L, LL, M, MCOUNT, NKEEP, NC, NCHNG, NTMP, NTMP1, N1
C
*COMDECK PEPT
      COMMON /PEPT/ ARGTST, EH, FSAVE1, FSAVE2, FTMP, FTMP1, FTNEW,
     1 FTNE1, PH, SAVE1, SAVE2, SAVE3, TMAX, TMIN, TNEW, XLTP1, XLTP2,
     2 XLTP3
C
C     COMMON /PEPT/ IS DESIGNED TO FAKE OUT THE FTN5 COMPILER IN OPT=2
C     MODE.  THESE VARIABLES ARE NEEDED ONLY IN SUBROUTINES PE AND PT,
C     BUT BY PLACING THEM IN A LABELLED COMMON WE FORCE THE COMPILER TO
C     RETAIN THEIR VALUES AFTER LAST USE FOR REUSE IN SUBSEQUENT
C     CALLS TO THE SUBROUTINE.
C
*COMDECK PLOTCM
      COMMON /PLTCM/ DSTF(10), MTLN(10), PSMAX(201), PSMIN(201), PX(201)
     1 , SQJ(10)
C
      COMMON /PLTCH/ MM(6)
      CHARACTER *10 MM
C
*COMDECK RZCOM
      COMMON /RZCOM/ RZC1, RZCO, RSCRIT
C
*COMDECK SPLLC
C     SPALL VARIABLES
C
      COMMON /SPLLC/ EM(6), IS, ISM, ISPALL, ISPLLM(6), JS, JTMAX, MS,
     1 SJ, SM(50), TMAX, TSPALL(201), US(50), XS(50)
C
```

HYPUF SOURCE LISTING

```
*DECK HIRAD
      PROGRAM HIRAD
C
*IF DEF,B32
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
*ENDIF
C
C                  *---*---*---*---*---*---*---*---*---*---*
C
C        THIS COMPUTER PROGRAM CALCULATES THE HYDRO OR ELASTIC-PLASTIC
C        MOTION OF UP TO SIX MATERIAL LAYERS.  EACH MATERIAL MAY INCLUDE
C        UP TO 5 ELEMENTS WITH THE TOTAL NUMBER OF ELEMENTS PER PROBLEM
C        LIMITED TO 10.
C
C        THE PROGRAM CAN ACCOMODATE UP TO 3 X-RAY SOURCES.
C
C        THE X-RAY ABSORPTION CROSS-SECTIONS OF THE ELEMENTS MAY BE
C        VARIABLE (DEPENDENT ON IONIZATION) IF DESIRED.
C
C        IF THE CROSS-SECTION OF AN ELEMENT IS VARIABLE THE PROGRAM
C        CALCULATES THE NUMBER OF X-RAY ABSORPTION EDGES THE ATOM
C        HAS AND THE ENERGY OF THESE EDGES.  IT ALSO CALCULATES THE
C        IONIZATION POTENTIALS FOR ZERO TO COMPLETE IONIZATION AND THE
C        IONIZATION ENERGY OF ALL THE POSSIBLE IONIZATION STATES.  IF IO
C        EQUALS 1, IT IS ASSUMED IN THESE CALCULATIONS THAT THE STABLE
C        CONFIGURATION OF THE ELECTRONS REMAINING AT ANY IONIZATION
C        STATE IS LIKE THAT OF A NEUTRAL ATOM CONTAINING AS MANY PROTONS
C        AS THE ION HAS ELECTRONS.   IF ION EQUALS ZERO, THE ELECTRON
C        CONFIGURATION REMAINS LIKE THAT OF THE ORIGINAL NEUTRAL ATOM
C        AND ELECTRONS ARE REMOVED FROM THE OUTSIDE SUB-SHELLS FIRST,
C        WORKING INWARD AS IONIZATION PROGRESSES.
C
C        EIGHT LEVELS OF IONIZATION (OR ALL POSSIBLE ONES IF Z IS LESS
C        THAN 8) ARE ALLOWED IN THE IONIZATION CALCULATIONS.
C
C        THE RELATIVE NUMBER OF PARTICLES IN THE LEVELS ARE DETERMINED
C        BY THE IONIZATION POTENTIALS OF THE LEVELS.
C
C        CONDUCTION MAY BE ALLOWED IN SOLID MATERIALS -- CONDUCTION AND
C        DIFFUSION MAY BE ALLOWED IN VAPORIZED MATERIALS.
C
C        THE EQUATION OF STATE SUBROUTINE OF THIS PROGRAM CALCULATES
C        NEW TEMPERATURES BY MAKING SUCCESSIVE GUESSES AT ALL ACTIVE
C        ZONE TEMPERATURES USING THE NEWTON METHOD.
C
C        THIS PROGRAM HAS BEEN SUPPLIED WITH A LINE PRINTER PLOTTING
C        ROUTINE WHICH OUTPUTS BOTH LOGARITHMIC AND LINEAR PLOTS WHEN
C        AND IF DESIRED FOR EASY AND QUICK INTERPRETATION OF PROBLEM
C        RESULTS.
```

```
C
C        DIAGNOSTIC PRINTOUTS OF IONIZATION CALCULATIONS, ENERGY
C        FLUXES, AND DEPOSITION OF ENERGY IN THE ZONES IS EASILY
C        CONTROLLED USING THE INPUT VARIABLES NPRIN AND JPRIN.
C
C        EVEN THOUGH THE PUFF-66 CODE, FROM WHICH HIRAD WAS DERIVED, DID
C        NOT CONFORM TO USASI FORTRAN PROGRAMMING STANDARDS, THE HIRAD
C        CODE HAS BEEN MADE TO COMPLY WITH THESE STANDARDS.  THIS HAS
C        FORCED THE ELIMINATION OF SCRATCH TAPE 6, SINCE THIS TAPE IS US
C        FOR PRINTED OUTPUT.  SCRATCH TAPE 4 WAS ELIMINATED AT THE SAME
C        TIME, SINCE THE EDITING PROGRAM USED FOR PUFF-66 WOULD NOT HAVE
C        BEEN DIRECTLY APPLICABLE TO HIRAD, ANYWAY.  IT IS FELT BY THE
C        AUTHORS THAT THE EDITING CAPABILITIES INCORPORATED IN HIRAD ARE
C        MORE THAN SUFFICIENT TO INFORM THE USER OF PROBLEM RESULTS, THU
C        MAKING TAPES 4 AND 6 NOT NECESSARY.
C
C                 *---*---*---*---*---*---*---*---*---*---*
C
C        VARIABLES IN BLANK COMMON AND THEIR USAGE
C
C        SUBSCRIPT -J- REFERS TO ZONE
C        SUBSCRIPT -M- REFERS TO MATERIAL (EITHER COMPOUND OR ELEMENT)
C        SUBSCRIPT -N- REFERS TO ELEMENT
C        SUBSCRIPT -L- REFERS TO ENERGY SOURCE
C        SUBSCRIPT -NS- REFERS TO SPECTRUM
C
C                       ZONE VARIABLES
C
C        CS(J)          SOUND SPEED IN CM/SECOND
C        V(J)           SPECIFIC VOLUME IN CC/GRAM
C        E(J)           SPECIFIC ENERGY IN ERGS/GRAM
C        P(J)           PRESSURE IN DYNES/CM**2
C        S(J)           STRESS IN DYNES/CM**2
C        SD(J)          STRESS DEVIATOR IN DYNES/CM**2
C        U(J)           PARTICLE VELOCITY IN CM/SECOND
C        ZM(J)          ZONE MASS IN GRAMS/CM**2
C        TEMP(J)        TEMPERATURE IN KELVIN DEGREES
C        ZFM(J)         IONIZATION IN MATERIAL
C        X(J)           ZONE BOUNDARY POSITION IN CM
C        Q(J)           ARTIFICIAL VISCOSITY FOR CURRENT TIME STEP IN
C                       DYNES/CM**2
C        QO(J)          ARTIFICIAL VISCOSITY FOR PREVIOUS TIME STEP
C        YOZ(J)         YIELD STRENGTH OF MATERIAL IN DYNES/CM**2 (IN
C                       THIS ZONE)
C        DV(J)          CHANGE IN SPECIFIC VOLUME DURING CURRENT TIME
C                       STEP
C        EI(J)          ENERGY USED IN IONIZATION IN THIS ZONE
C        ITER(J)        A FLAG SET UP IN SUBROUTINE EQST TO INDICATE TO
C                       SUBROUTINE FLOION NO CALCULATION OF CONDUCTION
C                       OR DIFFUSION FLUX INTO THIS ZONE IF ITER EQUALS
```

```
C                         ZERO, A CONDUCTION ONLY CALCULATION IF ITER
C                         EQUALS 1, A DIFFUSION ONLY CALCULATION IF ITER
C                         EQUALS 2, AND A DIFFUSION AND CONDUCTION
C                         CALCULATION IF ITER EQUALS 3.
C                         IF ITER EQUALS 4 NO ENERGY TRANSFER IS ALLOWED
C       F(J)              THE ENERGY FLUX INTO A ZONE IN ERGS/CM**2/SECOND
C                         DURING THE CURRENT TIME STEP
C       FO(J)             THE ENERGY FLUX CALCULATED DURING THE PREVIOUS
C                         CYCLE
C       EADD(J)           THE ENERGY ADDED TO THIS ZONE DURING THE
C                         CURRENT TIME STEP IN ERGS/GRAM
C       ZF(N,J)           THE IONIZATION IN ELEMENT N IN ZONE J DURING
C                         THE CURRENT TIME STEP
C       SS(J,L)           THE CURRENT VALUE OF THE ENERGY DEPOSITION
C                         RATE IN ERGS/GRAM/SECOND FOR ZONE J DUE TO
C                         SOURCE L
C       XFX(J)            X-RAY FLUX (CAL/SQ CM/SEC)
C       XFL(J)            X-RAY FLUENCE (CAL/SQ CM)
C
C       MATERIAL VARIABLES
C
C       AMU(M)            THE SHEAR MODULUS OR MODULUS OF RIGIDITY
C       YMU(M)            THE COMPRESSION WHICH CORRESPONDS TO THE ELASTIC
C                         LIMIT, I.E. PLASTIC FLOW OCCURS BEYOND THIS
C                         COMPRESSION.
C       YADD(M)           THE INCREASE IN YIELD STRENGTH WHICH OCCURS
C                         DURING COMPRESSION FROM YMU(M) TO 0.2
C       YO(M)             THE ROOM TEMPERATURE ZERO COMPRESSION YIELD
C                         STRENGTH
C
C       EQSTC(M)          SOLID EQUATION OF STATE
C       EQSTD(M)          CONSTANTS USED IN THE EQST --
C       EQSTS(M)          P=(C*MU+D*MU**2+S*MU**3)*(1.-G*MU/2.)+G*RHO*E
C       EQSTG(M)          WHERE MU = (RHO/RHOO)-1.
C
C       CUSP1(M)          THE VALUE OF THE PRESSURE AT WHICH AN INFLECTION
C                         OCCURS IN THE HUGONIOT DATA
C       CUSPA(M)          THE VALUE OF MU=(RHO/RHOO)-1. AT WHICH THE
C                         INFLECTION OCCURS
C
C       CUSPC(M)          SOLID EQUATION OF STATE CONSTANTS USED AT
C       CUSPD(M)          COMPRESSIONS BEYOND THE INFLECTION POINT IN
C       CUSPS(M)          THE EQST -- P=(CUSP1+CUSPC*(MU-CUSPA)+CUSPD*
C       CUSPG(M)          (MU-CUSPA)**2+CUSPS*(MU-CUSPA)**3)*(1.-CUSPG*
C                         MU/2.)+CUSPG*RHO*E
C
C       EQSTH(M)          VAPOR PHASE EQUATION OF STATE CONSTANTS USED
C       EQSTN(M)          IN THE EQST -- P=RHO*(EQSTH+(EQSTG-EQSTH)*
C       EQSTE(M)          ETA**0.5)*(E-EQSTE*(1.-EXP(EQSTN/ETA*(1.-
C                         1./ETA))))    WHERE ETA=RHO/RHOO AND EQSTE IS
```

```
C                       SUBLIMATION ENERGY, EQSTH IS SPECIFIC HEAT
C                       RATIO - 1, EQSTN IS EQSTC/(EQSTG*EQSTE*RHO)
C
C        MATL(M)        THE NAME OF THE MATERIAL
C        RHO(M)         THE ZERO-PRESSURE DENSITY
C        NELEM(M)       THE NUMGER OF ELEMENTS OCCURING IN THE MATERIAL
C        NATOM(M)       THE NUMBER OF ATOMS IN A MOLECULE OF THIS MATL.
C        XMW(M)         THE MOLECULAR WEIGHT OF THIS MATERIAL
C        LGDEL(M)       A MATERIAL CONSTANT USED IN CALCULATING THE
C                       CONDUCTIVITY AS -- CONDUCTIVITY=1.27E-5/
C                       AVERAGEIONIZATION/LGDEL*TEMPERATURE**2.5
C                       (ERGS/DEGREE/CM/SECOND)
C        XCON(M)        THE THERMAL CONDUCTIVITY OF THE NEUTRAL ATOMIC
C                       CONFIGURATION OF THIS MATERIAL
C        PMIN(M)        THE SMALLEST PRESSURE (LARGEST TENSION) ALLOWED
C                       IN A MATERIAL
C        JBND(M)        THE LAST ZONE IN A MATERIAL EXCEPT IF IT IS THE
C                       RIGHT MOST MATERIAL, WHEN JBND = 0
C        IELEM(M,N)     THE NUMBER DESIGNATION OF THE ELEMENTS (N)
C                       OCCURING IN MATERIAL M
C        AF(M,N)        THE ATOM FRACTION OF THE ELEMENTS IN THIS MATL.
C
C        ELEMENT VARIABLES
C
C        NAMEL(N)       ELEMENT NAME
C        XAW(N)         ATOMIC WEIGHT
C        NTBL(N)        ATOMIC NUMBER
C        NOE(N)         NUMBER OF X-RAY ABSORPTION EDGES
C        XI(N,100)      THE IONIZATION POTENTIALS
C        EN(N,100)      IONIZATION ENERGIES
C
C        AA(N,25)       ELEMENT CONSTANTS USED IN CALCULATING X-RAY
C        B(N,25)        ABSORPTION -- ABSORPTION COEFFICIENT= AA*
C                       PHOTON ENERGY**B IF PHOTON ENERGY GREATER THAN
C                       EDGE.
C
C        SOURCE VARIABLES
C
C        NBB(NS)        NUMBER OF SOURCES IN THIS SPECTRUM (AS MANY AS
C                       3 BLACKBODIES OR ONE ARBITRARY SOURCE)
C        T(L)           THE BLACKBODY TEMPERATURE (=1. FOR ARBITRARY
C                       SOURCE)
C        EE(L)          THE STRENGTH (CAL/CM**2)
C        NHNU(L)        NUMBER OF ENERGY INTERVALS (SET TO ZERO FOR
C                       BLACKBODY)
C        START(NS)      TIME OF BEGINNING OF SHINE
C        SSTOP(NS)      TIME OF END OF SHINE
C        ES(L,I)        ENERGY REMAINING IN ENERGY INTERVAL I OF
C                       SPECTRUM NS OR BLACKBODY L (USED IN CALCULATION
C                       OF ENERGY DEPOSITION)
```

```
C
C        ZONING VARIABLES
C
C        RZ(A)          GEOMETRIC RATIO OF ZONE SIZES
C        NZ(A)          NUMBER OF THE LAST ZONE USING RATIO RZ(A)
C
C        PROBLEM CONTROL, OUTPUT CONTROL, AND MISCELLANEOUS VARIABLES
C
C        DISCPT         PROBLEM DESCRIPTION
C        JEDIT          ZONE NUMBER WHERE LAGRANGIAN EDIT IS DESIRED
C        JORG           NOT USED
C        TEDIT          PROBLEM TIME IN SECONDS WHERE EDITS ARE DESIRED
C        CKS            WHEN PEAK PRESSURE REACHES THIS DEPTH, PROBLEM
C                       IS STOPPED.
C
C        CO             1.8
C        C1             0.25
C                       USED IN CALCULATION OF ARTIFICIAL VISCOSITY --
C                       QNEW=(DU**2*CO**2-C1*CS*DU)*RHO(AVE)   WHERE
C                       RHO(AVE) = 0.5*(RHO(OLD)+RHO(NEW))
C
C        DTN            0.5*(DTNH(OLD)+DTNH(NEW))
C        DTNH           CURRENT TIME STEP (SECONDS)
C        IT             VARIABLE USED TO CALL TIME EDITS
C        JCYCS          MAXIMUM NUMBER OF CYCLES IN PROBLEM
C        JFIN           ZONE BOUNDARY NUMBER OF LAST ZONE IN PROBLEM
C        JSMAX          THE NUMBER OF THE ZONE WITH THE LARGEST STRESS
C        JSMAXI         USED TO DETERMINE IF REZONE IS DESIRABLE
C        JRZL           ZONES COMBINED IF MORE THAN JRZL MEET
C                       QUALIFICATIONS
C        JSTAR          THE DEEPEST CURRENTLY ACTIVE ZONE
C        JTS            THE ZONE CONTROLLING THE TIME STEP
C        JZPUL          NUMBER OF ZONES DESIRED IN FRONT OF PRESSURE
C                       PULSE
C        LINE           NUMBER OF LINES PRINTED ON CURRENT OUTPUT
C                       RECORD -- USED TO CALL NEW PAGE.
C        LOZHIZ         A FLAG SET TO ONE FOR A TWO-PULSE PROGLEM
C        N              CURRENT HYDRO-CYCLE NUMBER
C        NJEDIT         NUMBER OF LAGRANGIAN EDITS
C        NMTRLS         NUMBER OF MATERIALS IN PROBLEM
C        NPRIN          EDITS ARE CALLED FOR EACH NPRIN HYDRO-CYCLES --
C                       ALSO, DIAGNOSTIC DATA IS PRINTED IF JPRIN=1
C        NREZON         REZONE CAN BE ENTERED ONLY ON MULTIPLES OF
C                       NREZON CYCLES
C        NRZ            THE ZONES IN THE DIVIDE AREA OF REZONE EXTEND TO
C                       JSMAX+NRZ
C        NSPEC          THE NUMBER OF SPECTRA IN THIS PROBLEM
C        NTAPE          DATA DUMPS ON TAPE OCCUR AT CYCLES WHICH ARE
C                       MULTIPLES OF NTAPE
C        NTEDIT         NUMBER OF TIME EDITS DESIRED
```

```
C        PDTNEG         INTEGRATED NEGATIVE MOMENTUM (DYNE-SECOND/CM**2)
C                       IN THE DEEPEST JEDIT ZONE
C        PDTPOS         INGEGRATED POSITIVE MOMENTUM IN DEEPEST JEDIT
C                       ZONE
C        SDURM          THE MINIMUM DURATION OF CURRENTLY ACTIVE SPECTRA
C                       -- USED IN CALCULATION OF MAXIMUM HYDRO TIME
C                       STEP ALLOWED
C        SK2M           MAXIMUM TIME STEP ALLOWED FOR STABILITY OF HYDRO
C                       MOTION
C        SMAX           MAXIMUM ABSOLUTE VALUE OF STRESS
C        SSTOPM         THE TIME AT WHICH ALL SOURCES HAVE STOPPED
C                       RADIATING
C        TIME           CURRENT PROBLEM TIME (SECONDS)
C        TS             INPUT STOP TIME FOR PROBLEM
C        WTAPE          A FLAG -- IF EQUAL 1, RESTART VARIABLES ARE
C                       WRITTEN ON TAPE
C        ILIN           IF 0, LINEAR PLOTS ARE MADE ON PRINTER EVERY
C                       NPRIN CYCLES
C        ILOG           IF ZERO, LOG PLOTS ARE MADE BY PRINTER EVERY
C                       NPRIN CYCLES
C        ICON           IF ZERO, CONDUCTIVITY IN  SOLID MATERIALS IS
C                       ASSUMED
C        IDIF           IF ZERO, DIFFUSION OF ENERGY IN MATERIALS IS
C                       ASSUMED
C        ANGLE          THE ANGLE IN STERADIANS AT WHICH THE SOURCES
C                       RADIATE ON THE MATERIAL
C        DTMIN          AN INPUT VARIABLE WHICH DETERMINES THE
C                       ACCURACY TO WHICH TEMPERATURES MUST BE
C                       CALCULATED BY THE EQST ROUTINE
C        DTPRIM         THE SMALLEST TIME OVER ALL ZONES IN WHICH
C                       THE ENERGY WILL BE ALTERED BY  1 PERCENT, DUE
C                       TO ENERGY TRANSFER WITH ITS NEIGHBORING MESHES.
C        IFLOW          A FLAG TO INDICATE TO THE FLOION SUBROUTINE THAT
C                       CONDUCTION AND/OR DIFFUSION RATES SHOULD BE
C                       RE-CALCULATED THIS CYCLE
C        JHAT           THE DEEPEST ZONE AT WHICH IONIZATION
C                       NEEDS TO BE CALCULATED
C        NCOUNT         A COUNT OF THE TOTAL NUMBER OF GUESSES REQUIRED
C                       BY THE EQST SUBROUTINE (ONE GUESS INCLUDES A NEW
C                       TEMPERATURE IN EACH ACTIVE ZONE) TO FIND
C                       CORRECT TEMPERATURES
C        NDEP           AN INPUT CONSTANT - ENERGY DEPOSITION IN THE
C                       ZONES IS RECALCULATED EVERY NDEP CYCLES
C        JPRIN          AN INPUT CONSTANT - SET EQUAL TO 1, IT CALLS
C                       FOR DIAGNOSTIC PRINTOUT AT MULTIPLES OF NPRIN
C                       CYCLES.  JPRIN CAN ALSO BE SET TO THE CYCLE
C                       NUMBER AT WHICH DELAYED DIAGNOSTIC PRINTOUT
C                       IS TO BEGIN.  IF JPRIN IS SET TO A NUMBER
C                       GREATER THAN ONE, DIAGNOSTIC OUTPUT FROM
C                       SUBROUTINE GENRAT IS PRINTED OUT AND THEN
```

```
C                       ONLY THE DIAGNOSTICS FOR NCYCL GREATER THAN
C                       JPRIN
C
C       ION             AN INPUT CONSTANT - SET EQUAL TO 1 IT CALLS FOR
C                       ELECTRON RE-ARRANGEMENT DURING IONIZATION.
C                       LEFT EQUAL TO ZERO ELECTRONS ARE REMOVED FROM
C                       OUTSIDE SUB-SHELLS WITH NO RE-ARRANGEMENT OF
C                       INNER SHELLS.
C
C                       VARIABLES IN COMMON BLOCK /AA/
C
C       ITBL(Z,6)       THESE SIX TABLES ARE USED TO FIND THE NORMAL
C       ILTBL1(3)       ELECTRON STRUCTURE OF THE ELEMENTS THROUGH
C       ILTBL2(3)       Z=96.  ITAB(Z,6) CONTAINS THE NUMBER OF
C       ILTBL3(3)       ELECTRONS IN THE LAST 1 TO 6 SUB-SHELL GROUPS -
C       ILTBL4(1)       S, P, D, AND F.  ILTBL1 THROUGH ILTBL5 CONTAIN
C       ILTBL5(3)       THE NUMBER OF ELECTRONS IN THE 1 TO 13 SUB-SHELL
C                       GROUPS.
C
C       ITABL(19)       THE 19 POSSIBLE SUB-SHELL GROUPS ARE COMPACTED
C       JTABL(14)       INTO 14 GROUPS IF THE S AND P SUB-SHELLS ARE
C                       COMBINED FOR CONVENIENCE IN CALCULATING
C                       SCREENING CONSTANTS AND ATOM OR ION ENERGIES.
C                       ITABL GIVES THE GROUP NUMBER IN THE COMPACTED
C                       GROUP (1 TO 14) INTO WHICH EACH OF THE
C                       SUB-SHELLS (1 TO 19) FITS.  JTABL GIVES THE
C                       HIGHEST NUMBER OF THE 1 TO 19 SUB-SHELL GROUPS
C                       INTO WHICH EACH OF THE COMPACTED GROUPS FITS.
C
C       XNSTAR(14)      THE QUANTUM SHELL NUMBER OF EACH OF THE
C                       COMPACTED GROUPS
C       TBL(109)        THE 99 ENERGIES AT WHICH EACH OF THE FIRST 99
C                       1 PERCENT INTERVALS OF ENERGY ARE CENTERED, AND
C                       THE 10 ENERGIES AT WHICH THE LAST 10 0.1 PERCENT
C                       INTERVALS OF ENERGY ARE CENTERED.
C       SCALE(18)       THE 18 SCALES AT WHICH LINEAR PRINTER PLOTS ARE
C                       MADE BY THE EDIT SUB-ROUTINE.
C       SCAX(15)        THE 15 DISTANCE SCALES (EACH DIVIDED INTO 100
C                       INTERVALS) USED FOR LINEAR PLOTTING IN EDIT.
C
C                       COMMON BLOCK /AB/
C
C       ICHCK           A FLAG SET UP IN GENRAT OR IN EQST TO INDICATE
C                       TO FLOION TO CALCULATE ZONE IONIZATION AND
C                       ENERGY TRANSFER IF ICHCK EQUALS 1, ZONE
C                       IONIZATION AND ENERGY DEPOSITION IF ICHCK
C                       EQUALS ZERO.
C
C                       COMMON BLOCK /AC/
C
```

```
C        NOEC(N)        THE NUMBER OF X-RAY ABSORPTION EDGES CALCULATED
C                       BY THE CODE TO EXIST FOR ELEMENT N -- THERE IS
C                       ONE FOR EACH OF THE S, P, D, AND F SUB-GROUPS
C                       IN THE ELEMENT.
C        EDGEC(N,20)    THE ENERGY CALCULATED BY THE CODE AT WHICH THE
C                       ABSORPTION EDGE OCCURS.
C        EION(N,100)    THE ENERGY OF ELEMENT N IONS WHEN 1 THROUGH ALL
C                       ELECTRONS ARE REMOVED FROM THE ATOM.
C        SCRENO(N,20)   THE SCREENING CONSTANTS ASSOCIATED WITH THE
C                       NORMAL    SUB-SHELL GROUPS FOR THE NORMAL ATOM
C                       OF ELEMENT N.
C        NGRUP(19)      THE NUMBER OF ELECTRONS IN THE SUB-SHELL GROUPS.
C        NSUM(19)       THE NUMBER OF ELECTRONS SUMMED FROM THE FIRST
C                       SUB-SHELL
C        NSPDF(14)      THE NUMBER OF ELECTRONS IN THE COMPACTED
C                       SUB-SHELL GROUPS
C        NVARM(M)       AN INPUT VARIABLE -- 1 DENOTES THAT ALL ELEMENTS
C                       IN THIS MATERIAL HAVE CONSTANT -COLD- ABSORPTION
C                       CROSS SECTIONS -- 0 DENOTES THAT THE ELEMENTS
C                       MAY HAVE VARIABLE CROSS SECTIONS.
C        NVARE(N)       IF THE MATERIAL IN WHICH THE ELEMENT OCCURS HAS
C                       NVARM = 0, AND NVARE = 1, THEN CROSS SECTION IS
C                       CONSTANT AND COLD, BUT IF NVARE = 0, THE CROSS
C                       SECTION IN THIS ELEMENT IS VARIABLE AND
C                       DEPENDENT ON IONIZATION IN THIS ELEMENT.
C        NION(N,19)     THE NUMBER OF ELECTRONS IN THE 19 SUB-SHELL
C                       GROUPS OF THE NORMAL ATOM OF THIS ELEMENT
C
C        COMMON BLOCK /EQVP/
C
C        VARIABLES USED FOR ELASTIC-VISCOPLASTIC AND GEOMETRIC DISPERSIO
C        MODELS.
C
C     AMU2(6)   EFFECTIVE SHEAR MODULUS FOR THE MAXWELL GEOMETRIC
C               DISPERSION MODEL.  USED WITH THE SECOND STRESS DEVIATOR
C               WHICH MODELS THE RATE DEPENDENT STRESS CONTRIBUTION.
C               INPUT VARIABLE.
C
C     CH(6)     CALCULATED IN GENRAT.  EQUAL TO THE SOUND SPEED IN THE
C               MATERIAL AT SOLID DENSITY.
C
C     EQSTA(6)  PARAMETER "A" IN THE DISPERSIVE MATERIAL MODEL FOR
C               VISCOPLASTIC BEHAVIOR.  INPUT VALUE.
C
C     GOKE(6)   CALCULATED IN GENRAT.  SHEAR MODULUS (AMU) DIVIDED BY
C               EQSTC
C
C     GOKE2(6)  CALCULATED IN GENRAT.  AMU2 DIVIDED BY EQSTC.
C
C     GOVERK(201)  INITIALIZED IN GENRAT AND REDEFINED IN HYDRO.
```

```
C              PLASTIC SHEAR STRAIN.
C
C     MFLAG(6)  FLAG USED IN HYDRO TO ACTIVAE THE MAXWELL GEOMETRIC
C              DISPERSION MODEL.  INPUT VALUE.
C
C     OMEGA(6)  PARAMETER "W" IN THE MAXWELL GEOMETRIC DISPERSION MODEL.
C              INPUT VALUE.
C
C     JB(6)    INDEX OF LEFT-HAND BOUNDARY OF FIRST ZONE TREATED AS
C              DISPERSIVE IN EACH DISPERSIVE MATERIAL.  CALCULATED
C              AND USED IN HYDRO.  ALSO USED IN EQST.
C
C     PRELAX(6)  RELAXATION STRESS USED IN VISCOPLASTIC MODEL.  INPUT
C              VALUE.
C
C     QU(201)  USED IN CALCULATION OF ARTIFICIAL VISCOSITY OF
C              DISPERSIVE MATERIALS.
C
C     SD2(201)  SECOND STRESS DEVIATOR USED IN MAXWELL GEOMETRIC
C              DISPERSION MODEL.  CALCULATED AND USED IN HYDRO.
C
C     SHEARR(6)  COEFFICIENT USED IN THE VARIABLE SHEAR MODULUS MODEL
C              ON THE UNLOADING PATH.  INPUT VALUE (SIMILAR TO
C              BURGER'S VECTOR)
C
C     TRELAX(6)  CHARACTERISTIC RELAXATION TIME FOR VISCOPLASTIC MODEL.
C              INPUT VALUE.
C
C     TRELX2(6)  RELAXATION TIME FOR THE MAXWELL GEOMETRIC DISPERSION
C              MODEL.  INPUT VALUE.
C
C     YY(201)  USED IN CALCULATION OF SD2 FOR DISPERSIVE MATERIALS.
C
C     ZZ(201)  USED IN CALCULATION OF SD2 FOR DISPERSIVE MATERIALS.
C
C           COMMON BLOCK /SPLLC/
C
C           VARIABLES USED TO CALCULATE FRACTURE
C
C           EM           MELT ENERGY OF THE MATERIAL.  INPUT QUANTITY.
C           IS           FLAG TO INDICATE IF A NEW SPALL HAS OCCURED
C                        THIS TIME STEP.
C           ISM         NUMBER OF SPALLS OCCURRING THIS TIME STEP.
C           ISPALL       FLAG TO INDICATE WHETHER FRACTURE IS TAKING
C                        PLACE. ZONES ARE BEING RECOMBINED, OR NOTHING
C                        TAKING PLACE AT ALL.
C           ISPLLM       FLAG TO INDICATE WHICH FRACTURE MODEL IS BEING
C                        USED FOR THE MATERIAL.  INPUT QUANTITY.
C                        = 1:  STRESS MODEL (BASED ON TENSILE STRESS)
C                        = 2:  STRAIN MODEL (BASED ON BULK TENSILE
C                              STRAIN)
C                        = 3-10:  RESERVED FOR MODELS YET TO BE INCLUI
C           JS           INDEX OF ZONE BEING SPALLED
C           MS           INDEX OF SPALL BOUNDARY
C           SJ           CALCULATED QUANTITY TO BE COMPARED WITH TSPALL
C                        TO DETERMINE IF SPALL HAS OCCURED
C           SM           MOMENTUM OF THE SPALLED MATERIAL
C           TSPALL       SPALL STRENGTH OF A ZONE BASED ON PMIN FOR THE
C                        MATERIAL IN THE SLID STATE AND WHETHER THE ZONE
C                        IS SOLID, LIQUID, GAS, OR PREVIOUSLY SPALLED.
C                        PMIN IS THE INPUT QUANTITY THAT INDICATES SPALL
C                        STRENGTH OF THE SOLID MATERIAL AND ITS VALUE IS
C                        BASED ON THE SPALL MODEL FOR THE MATERIAL.
C           XS           POSITION OF THE RIGHT HAND BOUNDARY OF THE
C                        SPALLED MATERIAL
C           US           VELOCITY OF THE RIGHT HAND BOUNDARY OF THE
C                        SPALLED MATERIAL
C
C                  *---*---*---*---*---*---*---*---*---*---*
C
C        IMPORTANT VARIABLES LOCAL TO THIS ROUTINE
C
C        TR             THE TIME AT WHICH ENERGY TRANSFER RATES SHOULD
C                       BE RECALCULATED BASED       ON        CHANGE
C                       OF ENERGY IN MESH (SEE TRPRIM)
C        TRPRIM         THE TIME AT WHICH THE ENERGY OF A ZONE WILL HAVE
C                       BEEN ALTERED BY 20 PERCENT DUE TO ENERGY
C                       TRANSFER ALONE
C
C                  *---*---*---*---*---*---*---*---*---*---*
C
*CALL BLANK
*CALL AA
*CALL AB
*CALL AC
*CALL EQED
*CALL EQFL
*CALL EQVP
*CALL HYEQ
*CALL PLOTCM
*CALL RZCOM
*CALL SPLLC
      REAL LGDEL
C
      CHARACTER *10 ZZCTRL
      DIMENSION ZZCTRL(40)
      DIMENSION ZIP(11366)
      EQUIVALENCE (ZIP(1),CS(1))
C
C     OPEN STATEMENTS FOR FORTRAN77
```

```
C
      OPEN (UNIT=1,ACCESS='SEQUENTIAL',FORM='UNFORMATTED',STATUS='SCR
     1H')
      OPEN (UNIT=2,ACCESS='SEQUENTIAL',FORM='UNFORMATTED',STATUS='NEW')
      OPEN (UNIT=3,STATUS='NEW')
      OPEN (UNIT=5,STATUS='OLD')
      OPEN (UNIT=6,STATUS='NEW')
      OPEN (UNIT=7,STATUS='NEW')
      OPEN (UNIT=8,STATUS='NEW')
      OPEN (UNIT=9,STATUS='NEW')
      OPEN (UNIT=10,ACCESS='SEQUENTIAL',FORM='UNFORMATTED',STATUS='NEW')
     1
C
C         ZEROES COMMON
C
      DO 10 J=1,11366
   10 ZIP(J)=0.
*IF DEF,B64
      CALL REMARK ('SSWITCH 1 WILL CAUSE PROGRAM TERMINATION')
*ENDIF
      CALL GENRAT
      N=1
      TR=SSTOPM
   20 SDURM=SSTOPM
      JTRY=1
   30 CALL HYDRO
      CALL EQST
C
C     NOW CHECK FOR SPALL
C
      CALL SPALL
      GO TO (40,30), JTRY
      CALL GOTOER
C
C         STOP PARAMETERS
C
   40 CONTINUE
C     WRITE (10) TIME,(XFX(J),J=1,JFIN)
*IF DEF,B64
C
C     CHECK TO SEE IF SSWITCH 1 IS SET.  IF SO, EXIT.
C
      CALL SSWTCH (1,J)
      IF (J.EQ.1) GO TO 90
*ENDIF
      IF (TIME.LE.SSTOPM) GO TO 110
      IF (SMAX) 50,100,50
   50 IF (TIME-TS) 60,90,90
   60 IF (N-JCYCS) 70,90,90
   70 IF (X(JSMAX)-CKS) 80,90,90
   80 CONTINUE
*IF DEF,B64
      CALL SSWTCH (2,J)
      IF (J.EQ.2) GO TO 110
      WRITE (6,360) N
*ENDIF
*IF DEF,B32
      GO TO 110
*ENDIF
   90 WTAPE=1.
      CALL EDIT
C     WRITE (6,330) (XFL(J),J=1,JFIN)
      IF (NJEDIT.NE.0) CALL PLOT
      STOP
  100 WRITE (6,350) N
      GO TO 90
C
C          EDIT CONTROLS
C
  110 CONTINUE
      IF (NJEDIT.NE.0) WRITE (8,330) TIME,(S(JEDIT(J)),J=1,NJEDIT)
      IF (JPRIN.EQ.1) GO TO 130
      IF (MOD(N,NTAPE)) 140,120,140
  120 WTAPE=0.
  130 CONTINUE
      IF (JPRIN.EQ.1) WTAPE=1.
*IF DEF,B64
      ENCODE (40,370,ZZCTRL) N,TIME,DTNH
C     CALLING OF REMARK PUTS A LINE OF OUTPUT DATA ON THE MACHINE CONSOL
C      SO THAT ONE MAY FOLLOW THE PROGRESS OF THE CALCULATION DURING
C      RUNNING OF THE PROBLEM IF DESIRED
C
      CALL REMARK (ZZCTRL)
*ENDIF
      CALL EDIT
      GO TO 160
  140 IF (MOD(N,NPRIN)) 160,150,160
  150 WTAPE=0.
      GO TO 130
C
  160 CONTINUE
C
C     NOW CHECK FOR REZONE
C
      IF (MOD(N,NREZON)) 180,170,180
  170 IF (TIME.GE.SSTOPM) CALL REZONE
  180 CONTINUE
C
C          TIME STEP CALCULATION
C
```

```
C        NEW TIME STEP (DTNH) IS THE MINIMUM OF THE STABLE HYDRO TIME
C        (SK2M), 1.1 TIMES THE OLD TIME STEP, AND THE STABLE TIME FOR
C        ENERGY TRANSFER FROM ZONE TO ZONE.
C
*IF DEF,B64
      IF (NTEDT.EQ.0) SK2M=AMIN1(0.9/SK2M,1.1*DTNH)
      IF (NTEDT.EQ.1) SK2M=AMIN1(0.9/SK2M,1.1*DTNKP)
*ENDIF
*IF DEF,B32
      IF (NTEDT.EQ.0) SK2M=DMIN1(0.9/SK2M,1.1*DTNH)
      IF (NTEDT.EQ.1) SK2M=DMIN1(0.9/SK2M,1.1*DTNKP)
*ENDIF
      IF (SSTOPM-TIME) 200,200,190
  190 CONTINUE
*IF DEF,B64
      SK2M=AMIN1(0.01*SDURM,SK2M)
*ENDIF
*IF DEF,B32
      SK2M=DMIN1(0.01*SDURM,SK2M)
*ENDIF
  200 DTN=DTNH
      DTNH=SK2M
      IF (IFLOW.EQ.0) GO TO 220
      IF (DTNH-DTPRIM/2.) 210,220,220
  210 IFLOW=0
      IF (TIME.LT.SSTOPM) IFLOW=1
*IF DEF,B64
      TR=TIME+AMIN1(DTPRIM,5.*DTNH)
*ENDIF
*IF DEF,B32
      TR=TIME+DMIN1(DTPRIM,5.*DTNH)
*ENDIF
C
C        TR    IS THE TIME AT WHICH THE ENERGY IN SOME ZONE WILL HAVE
C        BEEN ALTERED BY  5 PERCENT DUE TO ENERGY TRANSFER -- LIMIT
C        THE TIME FOR RECLACULATION OF ENERGY TRANSFER RATES TO THIS
C        TIME OR LESS.
C
C        TIME EDIT
C
  220 IF (NTEDT) 260,260,230
  230 WTAPE=1.
      CALL EDIT
      IT=IT+1
      IF (IT-26) 250,240,250
  240 IT=1
      TEDIT(1)=0.
  250 NTEDT=0
  260 IF (TEDIT(IT)) 290,290,270
  270 IF (TIME+DTNH-TEDIT(IT)) 290,280,280
  280 DTNKP=DTNH
      DTNH=TEDIT(IT)-TIME
      NTEDT=1
C
C        ADVANCE PROBLEM TIME
C
  290 TIME=TIME+DTNH
      DTN=DTN+DTNH
      DO 300 J=2,JFIN
  300 XFL(J)=XFL(J)+DTNH*XFX(J)
      IF (IDIF.NE.0.AND.ICON.NE.0) GO TO 310
C
C        IFLOW EQUALS ZERO IS A CALL FOR RECALCULATION OF ENERGY
C        TRANSFER RATES.
C
      IF (TIME.GT.TR) IFLOW=1
C
C        ADVANCE CYCLE NUMBER
C
  310 N=N+1
      IF (N.GE.NDBG) JPRIN=1
      IF (DTNH) 320,320,20
  320 WRITE (6,340) N
      STOP
C
  330 FORMAT (3X,1P11E12.4/)
  340 FORMAT (//,2X,15HDTNH=0 AT CYCLE,I5)
  350 FORMAT (//,2X,15HSMAX=0 AT CYCLE,I5)
  360 FORMAT (31H SENSE SWITCH 2 IS ON AT CYCLE ,I10)
  370 FORMAT (6HCYCLE=,I5,1X,2HT=,E11.3,1X,3HDT=,E11.3)
      END
```

```
*DECK AZONE
      SUBROUTINE AZONE (DX,NRZC)
*IF DEF,B32
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
*ENDIF
C
*CALL BLANK
*CALL PLOTCM
      DIMENSION A(20), C(20), D(20)
C
C     THE VALUE OF NRZC INDICATES WHETHER THIS IS AN INITIAL ZONING
C     ESTIMATE OR A REVISED ZONING CALCULATION
C
      IF (NRZC.LT.0) GO TO 50
C
C     THIS IS OUR INITIAL ESTIMATE.  NRZC IS ZERO AND RZ IS THE
C     THICKNESS OF EACH MATERIAL LAYER IN CM.  FIRST WE STORE THE
C     MATERIAL LAYER SIZES AND MASSES FOR LATER USE.
C
      DX=0.
      DO 10 I=1,NMTRLS
      A(I)=0.
      D(I)=RHO(I)*RZ(I)
      C(I)=RZ(I)
      J=I-1
      IF (I.GE.2) C(I)=C(I)+C(J)
   10 CONTINUE
      IF (NJEDIT.LE.0) GO TO 30
      DO 20 I=1,NJEDIT
      J=MTLN(I)
      K=J-1
      JEDIT(I)=I
      A(I)=DSTF(I)*C(J)
      IF (J.EQ.1) GO TO 20
      A(I)=C(K)+DSTF(I)*(C(J)-C(K))
   20 CONTINUE
C
      WRITE (6,170) (JEDIT(I),A(I),I=1,NJEDIT)
C
   30 CONTINUE
C
C     D IS NOW THE TOTAL MASS IN EACH MATERIAL LAYER AND C(NMTRLS)
C     IS THE TOTAL THICKNESS OF THE PROBLEM.
C
      N=150/NMTRLS
      M=N-1
      DX=1.E-5
      DO 40 I=1,NMTRLS
      IF (D(I).LE.0.) STOP 2
      NRZC=NRZC+N
      JBND(I)=NRZC+1
      J=2*I
      K=J-1
      NZ(K)=JBND(I)
      NZ(J)=JBND(I)+1
      D1=DX
      D2=D(I)
      CALL FINDRZ (D1,D2,N,RZ1)
      RZ(K)=RZ1
      D2=DX*RZ1**M
      L=I+1
      IF (I.EQ.NMTRLS) L=I
      RZ(J)=DX*RHO(L)/(D2*RHO(I))
      IF (JPRIN.EQ.1) WRITE (6,180) I,K,NZ(K),RZ(K),V(K),ZM(K)
C
   40 CONTINUE
C
      NRZC=0
      JFIN=JBND(NMTRLS)
C
C     WE HAVE COMPLETED OUR FIRST GUESS AT THE ZONING.  WE NOW REDEFINE
C     DX TO BE THE FIRST ZONE SIZE DIVIDED BY ZONE RATIO RATHER THAN
C     ZONE MASS.
C
      DX=DX/RHO(1)/RZ(1)
      RETURN
   50 CONTINUE
C
C    THIS IS OUR SECOND AND FINAL CALCULATION OF THE ZONING. DX HAS
C    BEEN REDEFINED AS THE DISTANCE THE STRESS WAVE WILL PROPOGATE
C    DURING THE SHINE TIME
C
      N=1
      D1=0.
      D2=0.
      DO 70 I=1,NMTRLS
      D2=D2+D(I)
      IF (DX.LT.C(I)) GO TO 60
      D1=D1+D(I)
      GO TO 70
   60 CONTINUE
      IF (N.EQ.1) D1=D1+D(I)+(DX-C(I))*RHO(I)
      N=N+1
   70 CONTINUE
C
C     WE HAVE TEMPORARILY DEFINED D1 AS THE MASS THROUGH WHICH WE
C     EXPECT THE SHOCK TO PROPOGATE DURING SHINE TIME.  WE HAVE ALSO
C     TEMPORARILY DEFINED D2 AS THE TOTAL MASS OF THE PROBLEM.  WE
C     NOW WISH TO REZONE THE PROBLEM WITH 50 EQUAL MASS ZONES FOR THE
```

```
C     REGION THROUGH WHICH THE SHOCK WILL PROPOGATE DURING SHINE TIME
C     AND 100 ZONES FOR THE REST OF THE PROBLEM.  HOWEVER, IF THE
C     SHOCK IS EXPECTED TO PROPOGATE TOO FAR, THEN WE WILL SIMPLY
C     LEAVE THE ZONING AS IT WAS IN OUR FIRST GUESS.
C
      IF (NMTRLS.EQ.1) C(2)=C(1)
      V(1)=X(2)
      K=1
      IF (DX.GE.C(2)) GO TO 130
      D3=D2-D1
      D4=.02*D1
      D5=.01*D3
C
C     NOW WE PLAN TO DEFINE NZ AS THE HIGHEST ZONE NUMBER IN A REGION,
C     RZ AS THE ZONE SIZE RATIO FOR THAT REGION, AND V AS THE
C     CORRESPONDING ZONE THICKNESSES IN THAT REGION.
C
      J=1
      K=0
      DO 120 I=1,NMTRLS
      IM1=I-1
      IF (DX.GT.C(I)) GO TO 110
C
C     DX IS .LE. C(I).  NEED TO SEE IF I .EQ. 1
C
      IF (I.GE.2) GO TO 90
C
C     NOW WE KNOW THAT DX .LE. C(1)
C
      K=K+1
      NZ(K)=51
      V(K)=.02*DX
      ZM(K)=D4
      RZ(K)=1.
      IF (JPRIN.EQ.1) WRITE (6,180) I,K,NZ(K),RZ(K),V(K),ZM(K)
      IF (DX.EQ.C(I)) GO TO 80
C
C     HERE DX .LT. C(1) .AND. DX .NE. C(1)
C
      N=K
      K=K+1
      ZM(K)=(C(I)-DX)*RHO(I)+ZM(N)
      NRZC=ZM(K)/D5
      NRZC=MAX0(NRZC,20)
      NZ(K)=NZ(N)+NRZC
      D1=D4
      D2=ZM(K)
      M=NRZC+1
      CALL FINDRZ (D1,D2,M,RZ1)
      RZ(K)=RZ1
      M=NRZC
      D2=D1*RZ1**M
      ZM(K)=D2
      V(K)=D2/RHO(I)
      IF (JPRIN.EQ.1) WRITE (6,180) I,K,NZ(K),RZ(K),V(K),ZM(K)
      JBND(I)=NZ(K)
      GO TO 120
   80 CONTINUE
C
C     HERE DX .EQ. C(1)
C
      JBND(I)=NZ(K)
      GO TO 120
   90 CONTINUE
C
C     NOW WE HAVE C(1) .LT. DX .LT. C(2)
C
      N=K
      K=K+1
      V(K)=DX-C(IM1)
      IF (V(K).LE.0.) GO TO 100
C
C     NOW WE ARE IN THE REGION BETWEEN C(1) AND DX.
C
      ZM(K)=V(K)*RHO(I)
      NRZC=ZM(K)/D4
      V(K)=V(K)/NRZC
      ZM(K)=V(K)*RHO(I)
      D1=ZM(K)
      RZ(K)=V(K)/V(N)
      NZ(K)=1+NZ(N)
      IF (JPRIN.EQ.1) WRITE (6,180) I,K,NZ(K),RZ(K),V(K),ZM(K)
      N=K
      K=K+1
      RZ(K)=1.
      NZ(K)=NZ(N)+NRZC-1
      ZM(K)=D1
      V(K)=D1/RHO(I)
      IF (JPRIN.EQ.1) WRITE (6,180) I,K,NZ(K),RZ(K),V(K),ZM(K)
C
C     NOW WE ARE IN THE REGION BETWEEN DX AND C(2)
C
      N=K
      K=K+1
      V(K)=C(I)-DX
      ZM(K)=V(K)*RHO(I)+D1
      NRZC=ZM(K)/D5
      NRZC=MAX0(NRZC,20)
      D2=ZM(K)
      M=NRZC+1
      CALL FINDRZ (D1,D2,M,RZ1)
      M=NRZC
      RZ(K)=RZ1
      D2=D1*RZ1**M
      NZ(K)=NZ(N)+NRZC
      ZM(K)=D2
      V(K)=D2/RHO(I)
      JBND(I)=NZ(K)
      IF (JPRIN.EQ.1) WRITE (6,180) I,K,NZ(K),RZ(K),V(K),ZM(K)
      GO TO 120
  100 CONTINUE
C
C     NOW WE ARE IN THE SITUATION WHERE DX .LT. C(IM1).  THEREFORE
C     WE SIMPLY ZONE FOR CONSTANT ZONE MASS RATIO.
C
      D1=ZM(N)
      D2=D(I)+D1
      NRZC=D2/D5
      NRZC=MAX0(NRZC,20)
      NZ(K)=NZ(N)+1
      RZ(K)=RHO(IM1)/RHO(I)
      V(K)=D1/RHO(I)
      IF (JPRIN.EQ.1) WRITE (6,180) I,K,NZ(K),RZ(K),V(K),ZM(K)
      N=K
      K=K+1
      M=NRZC
      CALL FINDRZ (D1,D2,M,RZ1)
      M=NRZC-1
      D2=D1*RZ1**M
      NZ(K)=NZ(N)+M-1
      RZ(K)=RZ1
      RZ(N)=RZ(N)*RZ1
      V(N)=RZ1*V(N)
      ZM(N)=RZ1*ZM(N)
      IF (JPRIN.EQ.1) WRITE (6,180) I,N,NZ(N),RZ(N),V(N),ZM(N)
      ZM(K)=D2
      V(K)=D2/RHO(I)
      IF (JPRIN.EQ.1) WRITE (6,180) I,K,NZ(K),RZ(K),V(K),ZM(K)
      JBND(I)=NZ(K)
      GO TO 120
  110 CONTINUE
C
C     WE ARE IN THE FIRST MATERIAL REGION AND C(1) .LT. DX.
C
      K=K+1
      ZM(K)=D(I)
      NRZC=ZM(K)/D4
      NZ(K)=NRZC+1
      RZ(K)=1.
      ZM(K)=ZM(K)/NRZC
      V(K)=ZM(K)/RHO(I)
      IF (JPRIN.EQ.1) WRITE (6,180) I,K,NZ(K),RZ(K),V(K),ZM(K)
      JBND(I)=NZ(K)
  120 CONTINUE
C
C    WE HAVE NOW FINISHED CALCULATING NZ AND RZ.  ALL WE NEED TO DO IS
C    SET DX = V(1) AND THEN ZERO OUT THE X, V, AND U ARRAYS.  FINALLY,
C    WE RESET NRZC = - K AS A FLAG TO SHOW WE HAVE FINISHED OUR
C    CALCULATION OF ZONE SIZES.
C
      X(2)=V(1)
  130 CONTINUE
      DX=X(2)
      D1=DX/RZ(1)
      JFIN=NZ(K)
      J=1
      K=1
      X(1)=0.
      DO 150 I=2,JFIN
      IM1=I-1
      V(I)=0.
      ZM(I)=0.
      IF (I.GT.NZ(K)) K=K+1
      D1=D1*RZ(K)
      X(I)=X(IM1)+D1
      IF (J.GT.NJEDIT) GO TO 150
  140 CONTINUE
      IF (X(I).LT.A(J)) GO TO 150
      JEDIT(J)=I
      J=J+1
      IF (J.LE.NJEDIT) GO TO 140
  150 CONTINUE
      IF (NJEDIT.EQ.0) GO TO 160
      WRITE (6,170) (JEDIT(I),X(JEDIT(I)),I=1,NJEDIT)
  160 CONTINUE
      V(1)=0.
      ZM(1)=0.
      NRZC=-K
      RETURN
C
C**  THIS PROGRAM VALID ON FTN4 AND FTN5 **
C
  170 FORMAT (20H0  JEDIT, X(JEDIT) =,5(I5,1PE12.4)/)
  180 FORMAT (1H0,3HI =,I3,5X,3HK =,I3,5X,4HNZ =,I4,5X,4HRZ =,1PE15.7,5X
     1 ,3HV =,1PE15.7,5X,4HZM =,1PE15.7/)
      END
```

```
*DECK BLKDAT
      BLOCK DATA
*IF DEF,B32
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
*ENDIF
C
*CALL AA
*CALL AB
*CALL AC
*CALL EQED
*CALL EQFL
*CALL EQVP
*CALL INDX
*CALL SPLLC
      DATA ITBL(1,1), ITBL(1,2), ITBL(1,3), ITBL(1,4), ITBL(1,5), ITBL(1
     1 ,6) /1,0,0,0,0,0/
      DATA ITBL(2,1), ITBL(2,2), ITBL(2,3), ITBL(2,4), ITBL(2,5), ITBL(2
     1 ,6) /2,0,0,0,0,0/
      DATA ITBL(3,1), ITBL(3,2), ITBL(3,3), ITBL(3,4), ITBL(3,5), ITBL(3
     1 ,6) /2,1,0,0,0,0/
      DATA ITBL(4,1), ITBL(4,2), ITBL(4,3), ITBL(4,4), ITBL(4,5), ITBL(4
     1 ,6) /2,2,0,0,0,0/
      DATA ITBL(5,1), ITBL(5,2), ITBL(5,3), ITBL(5,4), ITBL(5,5), ITBL(5
     1 ,6) /2,2,1,0,0,0/
      DATA ITBL(6,1), ITBL(6,2), ITBL(6,3), ITBL(6,4), ITBL(6,5), ITBL(6
     1 ,6) /2,2,2,0,0,0/
      DATA ITBL(7,1), ITBL(7,2), ITBL(7,3), ITBL(7,4), ITBL(7,5), ITBL(7
     1 ,6) /2,2,3,0,0,0/
      DATA ITBL(8,1), ITBL(8,2), ITBL(8,3), ITBL(8,4), ITBL(8,5), ITBL(8
     1 ,6) /2,2,4,0,0,0/
      DATA ITBL(9,1), ITBL(9,2), ITBL(9,3), ITBL(9,4), ITBL(9,5), ITBL(9
     1 ,6) /2,2,5,0,0,0/
      DATA ITBL(10,1), ITBL(10,2), ITBL(10,3), ITBL(10,4), ITBL(10,5),
     1 ITBL(10,6) /2,2,6,0,0,0/
      DATA ILTBL1 /2,2,6/
C
      DATA ITBL(11,1), ITBL(11,2), ITBL(11,3), ITBL(11,4), ITBL(11,5),
     1 ITBL(11,6) /1,0,0,0,0,0/
      DATA ITBL(12,1), ITBL(12,2), ITBL(12,3), ITBL(12,4), ITBL(12,5),
     1 ITBL(12,6) /2,0,0,0,0,0/
      DATA ITBL(13,1), ITBL(13,2), ITBL(13,3), ITBL(13,4), ITBL(13,5),
     1 ITBL(13,6) /2,1,0,0,0,0/
      DATA ITBL(14,1), ITBL(14,2), ITBL(14,3), ITBL(14,4), ITBL(14,5),
     1 ITBL(14,6) /2,2,0,0,0,0/
      DATA ITBL(15,1), ITBL(15,2), ITBL(15,3), ITBL(15,4), ITBL(15,5),
     1 ITBL(15,6) /2,3,0,0,0,0/
      DATA ITBL(16,1), ITBL(16,2), ITBL(16,3), ITBL(16,4), ITBL(16,5),
     1 ITBL(16,6) /2,4,0,0,0,0/
      DATA ITBL(17,1), ITBL(17,2), ITBL(17,3), ITBL(17,4), ITBL(17,5),
     1 ITBL(17,6) /2,5,0,0,0,0/
      DATA ITBL(18,1), ITBL(18,2), ITBL(18,3), ITBL(18,4), ITBL(18,5),
     1 ITBL(18,6) /2,6,0,0,0,0/
      DATA ITBL(19,1), ITBL(19,2), ITBL(19,3), ITBL(19,4), ITBL(19,5),
     1 ITBL(19,6) /2,6,0,1,0,0/
      DATA ITBL(20,1), ITBL(20,2), ITBL(20,3), ITBL(20,4), ITBL(20,5),
     1 ITBL(20,6) /2,6,0,2,0,0/
      DATA ITBL(21,1), ITBL(21,2), ITBL(21,3), ITBL(21,4), ITBL(21,5),
     1 ITBL(21,6) /2,6,1,2,0,0/
      DATA ITBL(22,1), ITBL(22,2), ITBL(22,3), ITBL(22,4), ITBL(22,5),
     1 ITBL(22,6) /2,6,2,2,0,0/
      DATA ITBL(23,1), ITBL(23,2), ITBL(23,3), ITBL(23,4), ITBL(23,5),
     1 ITBL(23,6) /2,6,3,2,0,0/
      DATA ITBL(24,1), ITBL(24,2), ITBL(24,3), ITBL(24,4), ITBL(24,5),
     1 ITBL(24,6) /2,6,5,1,0,0/
      DATA ITBL(25,1), ITBL(25,2), ITBL(25,3), ITBL(25,4), ITBL(25,5),
     1 ITBL(25,6) /2,6,5,2,0,0/
      DATA ITBL(26,1), ITBL(26,2), ITBL(26,3), ITBL(26,4), ITBL(26,5),
     1 ITBL(26,6) /2,6,6,2,0,0/
      DATA ITBL(27,1), ITBL(27,2), ITBL(27,3), ITBL(27,4), ITBL(27,5),
     1 ITBL(27,6) /2,6,7,2,0,0/
      DATA ITBL(28,1), ITBL(28,2), ITBL(28,3), ITBL(28,4), ITBL(28,5),
     1 ITBL(28,6) /2,6,8,2,0,0/
      DATA ILTBL2 /2,6,10/
C
      DATA ITBL(29,1), ITBL(29,2), ITBL(29,3), ITBL(29,4), ITBL(29,5),
     1 ITBL(29,6) /1,0,0,0,0,0/
      DATA ITBL(30,1), ITBL(30,2), ITBL(30,3), ITBL(30,4), ITBL(30,5),
     1 ITBL(30,6) /2,0,0,0,0,0/
      DATA ITBL(31,1), ITBL(31,2), ITBL(31,3), ITBL(31,4), ITBL(31,5),
     1 ITBL(31,6) /2,1,0,0,0,0/
      DATA ITBL(32,1), ITBL(32,2), ITBL(32,3), ITBL(32,4), ITBL(32,5),
     1 ITBL(32,6) /2,2,0,0,0,0/
      DATA ITBL(33,1), ITBL(33,2), ITBL(33,3), ITBL(33,4), ITBL(33,5),
     1 ITBL(33,6) /2,3,0,0,0,0/
      DATA ITBL(34,1), ITBL(34,2), ITBL(34,3), ITBL(34,4), ITBL(34,5),
     1 ITBL(34,6) /2,4,0,0,0,0/
      DATA ITBL(35,1), ITBL(35,2), ITBL(35,3), ITBL(35,4), ITBL(35,5),
     1 ITBL(35,6) /2,5,0,0,0,0/
      DATA ITBL(36,1), ITBL(36,2), ITBL(36,3), ITBL(36,4), ITBL(36,5),
     1 ITBL(36,6) /2,6,0,0,0,0/
      DATA ITBL(37,1), ITBL(37,2), ITBL(37,3), ITBL(37,4), ITBL(37,5),
     1 ITBL(37,6) /2,6,0,0,1,0/
      DATA ITBL(38,1), ITBL(38,2), ITBL(38,3), ITBL(38,4), ITBL(38,5),
     1 ITBL(38,6) /2,6,0,0,2,0/
      DATA ITBL(39,1), ITBL(39,2), ITBL(39,3), ITBL(39,4), ITBL(39,5),
     1 ITBL(39,6) /2,6,1,0,2,0/
      DATA ITBL(40,1), ITBL(40,2), ITBL(40,3), ITBL(40,4), ITBL(40,5),
     1 ITBL(40,6) /2,6,2,0,2,0/
      DATA ITBL(41,1), ITBL(41,2), ITBL(41,3), ITBL(41,4), ITBL(41,5),
     1 ITBL(41,6) /2,6,4,0,1,0/
      DATA ITBL(42,1), ITBL(42,2), ITBL(42,3), ITBL(42,4), ITBL(42,5),
     1 ITBL(42,6) /2,6,5,0,1,0/
      DATA ITBL(43,1), ITBL(43,2), ITBL(43,3), ITBL(43,4), ITBL(43,5),
     1 ITBL(43,6) /2,6,6,0,1,0/
      DATA ITBL(44,1), ITBL(44,2), ITBL(44,3), ITBL(44,4), ITBL(44,5),
     1 ITBL(44,6) /2,6,7,0,1,0/
      DATA ITBL(45,1), ITBL(45,2), ITBL(45,3), ITBL(45,4), ITBL(45,5),
     1 ITBL(45,6) /2,6,8,0,1,0/
      DATA ITBL(46,1), ITBL(46,2), ITBL(46,3), ITBL(46,4), ITBL(46,5),
     1 ITBL(46,6) /2,6,10,0,0,0/
      DATA ILTBL3 /2,6,10/
C
      DATA ITBL(47,1), ITBL(47,2), ITBL(47,3), ITBL(47,4), ITBL(47,5),
     1 ITBL(47,6) /0,1,0,0,0,0/
      DATA ITBL(48,1), ITBL(48,2), ITBL(48,3), ITBL(48,4), ITBL(48,5),
     1 ITBL(48,6) /0,2,0,0,0,0/
      DATA ITBL(49,1), ITBL(49,2), ITBL(49,3), ITBL(49,4), ITBL(49,5),
     1 ITBL(49,6) /0,2,1,0,0,0/
      DATA ITBL(50,1), ITBL(50,2), ITBL(50,3), ITBL(50,4), ITBL(50,5),
     1 ITBL(50,6) /0,2,2,0,0,0/
      DATA ITBL(51,1), ITBL(51,2), ITBL(51,3), ITBL(51,4), ITBL(51,5),
     1 ITBL(51,6) /0,2,3,0,0,0/
      DATA ITBL(52,1), ITBL(52,2), ITBL(52,3), ITBL(52,4), ITBL(52,5),
     1 ITBL(52,6) /0,2,4,0,0,0/
      DATA ITBL(53,1), ITBL(53,2), ITBL(53,3), ITBL(53,4), ITBL(53,5),
     1 ITBL(53,6) /0,2,5,0,0,0/
      DATA ITBL(54,1), ITBL(54,2), ITBL(54,3), ITBL(54,4), ITBL(54,5),
     1 ITBL(54,6) /0,2,6,0,0,0/
      DATA ITBL(55,1), ITBL(55,2), ITBL(55,3), ITBL(55,4), ITBL(55,5),
     1 ITBL(55,6) /0,2,6,0,0,1/
      DATA ITBL(56,1), ITBL(56,2), ITBL(56,3), ITBL(56,4), ITBL(56,5),
     1 ITBL(56,6) /0,2,6,0,0,2/
      DATA ITBL(57,1), ITBL(57,2), ITBL(57,3), ITBL(57,4), ITBL(57,5),
     1 ITBL(57,6) /0,2,6,1,0,2/
      DATA ITBL(58,1), ITBL(58,2), ITBL(58,3), ITBL(58,4), ITBL(58,5),
     1 ITBL(58,6) /2,2,6,0,0,2/
      DATA ITBL(59,1), ITBL(59,2), ITBL(59,3), ITBL(59,4), ITBL(59,5),
     1 ITBL(59,6) /3,2,6,0,0,2/
      DATA ITBL(60,1), ITBL(60,2), ITBL(60,3), ITBL(60,4), ITBL(60,5),
     1 ITBL(60,6) /4,2,6,0,0,2/
      DATA ITBL(61,1), ITBL(61,2), ITBL(61,3), ITBL(61,4), ITBL(61,5),
     1 ITBL(61,6) /5,2,6,0,0,2/
      DATA ITBL(62,1), ITBL(62,2), ITBL(62,3), ITBL(62,4), ITBL(62,5),
     1 ITBL(62,6) /6,2,6,0,0,2/
      DATA ITBL(63,1), ITBL(63,2), ITBL(63,3), ITBL(63,4), ITBL(63,5),
     1 ITBL(63,6) /7,2,6,0,0,2/
      DATA ITBL(64,1), ITBL(64,2), ITBL(64,3), ITBL(64,4), ITBL(64,5),
     1 ITBL(64,6) /7,2,6,1,0,2/
      DATA ITBL(65,1), ITBL(65,2), ITBL(65,3), ITBL(65,4), ITBL(65,5),
     1 ITBL(65,6) /9,2,6,0,0,2/
      DATA ITBL(66,1), ITBL(66,2), ITBL(66,3), ITBL(66,4), ITBL(66,5),
     1 ITBL(66,6) /10,2,6,0,0,2/
      DATA ITBL(67,1), ITBL(67,2), ITBL(67,3), ITBL(67,4), ITBL(67,5),
     1 ITBL(67,6) /11,2,6,0,0,2/
      DATA ITBL(68,1), ITBL(68,2), ITBL(68,3), ITBL(68,4), ITBL(68,5),
     1 ITBL(68,6) /12,2,6,0,0,2/
      DATA ITBL(69,1), ITBL(69,2), ITBL(69,3), ITBL(69,4), ITBL(69,5),
     1 ITBL(69,6) /13,2,6,0,0,2/
      DATA ILTBL4(1) /14/
C
      DATA ITBL(70,1), ITBL(70,2), ITBL(70,3), ITBL(70,4), ITBL(70,5),
     1 ITBL(70,6) /2,6,0,0,2,0/
      DATA ITBL(71,1), ITBL(71,2), ITBL(71,3), ITBL(71,4), ITBL(71,5),
     1 ITBL(71,6) /2,6,1,0,2,0/
      DATA ITBL(72,1), ITBL(72,2), ITBL(72,3), ITBL(72,4), ITBL(72,5),
     1 ITBL(72,6) /2,6,2,0,2,0/
      DATA ITBL(73,1), ITBL(73,2), ITBL(73,3), ITBL(73,4), ITBL(73,5),
     1 ITBL(73,6) /2,6,3,0,2,0/
      DATA ITBL(74,1), ITBL(74,2), ITBL(74,3), ITBL(74,4), ITBL(74,5),
     1 ITBL(74,6) /2,6,4,0,2,0/
      DATA ITBL(75,1), ITBL(75,2), ITBL(75,3), ITBL(75,4), ITBL(75,5),
     1 ITBL(75,6) /2,6,5,0,2,0/
      DATA ITBL(76,1), ITBL(76,2), ITBL(76,3), ITBL(76,4), ITBL(76,5),
     1 ITBL(76,6) /2,6,6,0,2,0/
      DATA ITBL(77,1), ITBL(77,2), ITBL(77,3), ITBL(77,4), ITBL(77,5),
     1 ITBL(77,6) /2,6,7,0,2,0/
      DATA ITBL(78,1), ITBL(78,2), ITBL(78,3), ITBL(78,4), ITBL(78,5),
     1 ITBL(78,6) /2,6,9,0,1,0/
      DATA ILTBL5 /2,6,10/
C
      DATA ITBL(79,1), ITBL(79,2), ITBL(79,3), ITBL(79,4), ITBL(79,5),
     1 ITBL(79,6) /0,1,0,0,0,0/
      DATA ITBL(80,1), ITBL(80,2), ITBL(80,3), ITBL(80,4), ITBL(80,5),
     1 ITBL(80,6) /0,2,0,0,0,0/
      DATA ITBL(81,1), ITBL(81,2), ITBL(81,3), ITBL(81,4), ITBL(81,5),
     1 ITBL(81,6) /0,2,1,0,0,0/
      DATA ITBL(82,1), ITBL(82,2), ITBL(82,3), ITBL(82,4), ITBL(82,5),
     1 ITBL(82,6) /0,2,2,0,0,0/
      DATA ITBL(83,1), ITBL(83,2), ITBL(83,3), ITBL(83,4), ITBL(83,5),
     1 ITBL(83,6) /0,2,3,0,0,0/
      DATA ITBL(84,1), ITBL(84,2), ITBL(84,3), ITBL(84,4), ITBL(84,5),
     1 ITBL(84,6) /0,2,4,0,0,0/
      DATA ITBL(85,1), ITBL(85,2), ITBL(85,3), ITBL(85,4), ITBL(85,5),
     1 ITBL(85,6) /0,2,5,0,0,0/
      DATA ITBL(86,1), ITBL(86,2), ITBL(86,3), ITBL(86,4), ITBL(86,5),
     1 ITBL(86,6) /0,2,6,0,0,0/
      DATA ITBL(87,1), ITBL(87,2), ITBL(87,3), ITBL(87,4), ITBL(87,5),
     1 ITBL(87,6) /0,2,6,0,0,1/
      DATA ITBL(88,1), ITBL(88,2), ITBL(88,3), ITBL(88,4), ITBL(88,5),
     1 ITBL(88,6) /0,2,6,0,0,2/
      DATA ITBL(89,1), ITBL(89,2), ITBL(89,3), ITBL(89,4), ITBL(89,5),
     1 ITBL(89,6) /0,2,6,1,0,2/
      DATA ITBL(90,1), ITBL(90,2), ITBL(90,3), ITBL(90,4), ITBL(90,5),
     1 ITBL(90,6) /0,2,6,2,0,2/
      DATA ITBL(91,1), ITBL(91,2), ITBL(91,3), ITBL(91,4), ITBL(91,5),
     1 ITBL(91,6) /2,2,6,1,0,2/
      DATA ITBL(92,1), ITBL(92,2), ITBL(92,3), ITBL(92,4), ITBL(92,5),
     1 ITBL(92,6) /3,2,6,1,0,2/
      DATA ITBL(93,1), ITBL(93,2), ITBL(93,3), ITBL(93,4), ITBL(93,5),
     1 ITBL(93,6) /4,2,6,1,0,2/
      DATA ITBL(94,1), ITBL(94,2), ITBL(94,3), ITBL(94,4), ITBL(94,5),
     1 ITBL(94,6) /5,2,6,1,0,2/
      DATA ITBL(95,1), ITBL(95,2), ITBL(95,3), ITBL(95,4), ITBL(95,5),
     1 ITBL(95,6) /6,2,6,1,0,2/
      DATA ITBL(96,1), ITBL(96,2), ITBL(96,3), ITBL(96,4), ITBL(96,5),
     1 ITBL(96,6) /7,2,6,1,0,2/
C
      DATA ITABL /1,2,2,3,3,4,5,5,6,7,8,8,9,10,11,11,12,13,14/
C
      DATA XNSTAR /1.0,4.0,2*9.0,3*13.69,3*16.0,3*17.64,19.36/
C
      DATA JTABL /1,2,4,6,7,9,10,11,13,14,15,17,18,19/
C
      DATA TBL /.49,.729,.883,1.004,1.108,1.199,1.283,1.36,1.433,1.502,1
     1 .573,1.630,1.690,1.749,1.806,1.861,1.916,1.968,2.020,2.071,2.133,
     2 2.176,2.220,2.268,2.316,2.3639,2.4111,2.4579,2.5046,2.5508,2.5970
     3 ,2.6430,2.6889,2.7346,2.7804,2.8261,2.8717,2.9175,2.9633,3.0091,3
     4 .0551,3.1012,3.1476,3.1940,3.2475,3.2877,3.3350,3.3803,3.4305,3.4
     5 788,3.5270,3.5770,3.6260,3.6760,3.7270,3.7780,3.8300,3.8820,3.935
     6 0,3.9890,4.0440,4.0990,4.1550,4.2130,4.2710,4.3300,4.391,4.453,4.
     7 516,4.580,4.646,4.714,4.783,4.855,4.928,5.004,5.088,5.160,5.245,5
     8 .332,5.420,5.516,5.614,5.716,5.824,5.939,6.058,6.188,6.326,6.475,
     9 6.637,6.815,7.012,7.235,7.491,7.792,8.166,8.654,9.377,9.677,9.977
     $ ,10.277,10.577,10.877,11.177,11.477,11.777,12.037,12.377/
C
      DATA SCAX /.001,.002,.005,.01,.02,.05,.1,.2,.5,1.,2.,5.,10.,20.,50
     1 ./
C
      DATA SCALE /2.5E9,5.E9,1.E10,2.5E10,5.E10,1.E11,2.5E11,5.E11,1.E12
     1 ,2.5E12,5.E12,1.E13,2.5E13,5.E13,1.E14,2.5E14,5.E14,1.E15/
C
      DATA NOEC, EDGEC, EION, SCRENO, NGRUP, NSUM, NSPDF, NVARM, NVARE,
     1 NION, EBB, NLEC, NSNIK /10*0,200*0,1000*0,200*0.,19*0,19*0,14*0,6
     2 *0,10*0,190*0,2100*0.,150*0,1120*0/
C
      DATA IPLUS, IPLUSO, KMAX, NI, R, XLAM1, XLAM2, ZF1 /10*0,10*0,10*0
     1 ,80*0,80*0.,201*0.,201*0.,10*0./
C
      DATA RERAD /0./
```

```
C
      DATA AMU2, CH, EQSTA, GOKE, GOKE2, GOVERK, JB, MFLAG, OMEGA,
     1 PRELAX, QU, SD2, SHEARR, TRELAX, TRELX2, VAMU, YY, ZZ /6*0.,6*C.,
     2 6*0.,6*0.,6*0.,201*0.,6*0,6*0,6*0.,6*0.,201*0.,201*C.,6*0.,6*C.,6
     3 *0.,201*0.,201*0.,201*0./
C
      DATA KCOUNT /0/
C
      DATA EM, IS, ISM, ISPALL, ISPLLM, JS, JTMAX, MS, SJ, SM, TMAX,
     1 TSPALL, US, XS /6*0.,0,0,0,6*0,0,0,0,0.,50*0.,0.,201*0.,50*0.,50*
     2 0./
C
C**  THIS PROGRAM VALID ON FTN4 AND FTN5 **
      END
```

```
*DECK EDIT
      SUBROUTINE EDIT
*IF DEF,B32
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
*ENDIF
C
C                  *---*---*---*---*---*---*---*---*---*---*
C
C        IMPORTANT VARIABLES LOCAL TO THIS SUBROUTINE
C
C        DTPP           PULSE WIDTH IN SECONDS AT HALF HEIGHT OF THE
C                       PULSE CONTAINING SMAX -- COMPUTED USING THE
C                       EXPRESSION   EMVPP/SMAX
C        DTPULS         A PULSE WIDTH TERM IN SECONDS, COMPUTED USING
C                       THE EXPRESSION  EMVPL/SMAX
C        EMVBM          THE SUM OF THE MAXIMUM POTENTIAL MOMENTUM IN
C                       DYNE-SECONDS/CM**2 OF VAPORIZED REGION --
C                       SUBLAYER REGIONS THAT ARE VAPORIZED ARE NOT
C                       CONSIDERED. -- THE ENERGY E AVALIABLE FOR THE
C                       MAXIMUM MOMENTUM OF THE ZONE IS DEFINED AS
C                       E=0.5*U**2+(E-ESUB) WHERE ESUB IS THE
C                       SUBLIMATION ENERGY -- THE VELOCITY U ASSOCIATED
C                       WITH THE MAXIMUM POTENTIAL MOMENTUM IS DEFINED
C                       AS E=0.5*U**2 OR U=SQRT(2*E) -- HENCE THE
C                       MAXIMUM POTENTIAL FOR A VAPORIZED ZONE IS
C                       ZM*SQRT(U**2+2*(E-ESUB))
C        EMVNEG         TOTAL NEGATIVE MOMENTUM IN DYNE-SECONDS/CM**2 IN
C                       A MESH -- THIS EXPRESSION IS COMPUTED BY
C                       AVERAGING THE VELOCITIES OF TWO ADJACENT ZONE
C                       BOUNDARIES AND USING THE MASS OF THE ZONE
C                       BETWEEN.
C        EMVPL          SUM OF THE MOMENTUM OF EACH ZONE FROM JSMAX+3
C                       BACK TO THE FIRST ZONE WHICH HAS A NEGATIVE
C                       VELOCITY.
C        EMVPOS         TOTAL POSITIVE MOMENTUM IN A MESH -- CALCULATED
C                       LIKE EMVNEG
C        EMVPP          SUM OF EMVPL AND EMVPR
C        EMVPR          SUM OF THE MOMENTUM OF EACH ZONE FROM JSMAX+4
C                       TO JSTAR+1
C        ETOTAL         TOTAL ENERGY IN CALORIES COMPUTED BY SUMMING THE
C                       KINETIC AND INTERNAL ENERGY OF ALL ZONES.
C        JQMAX          ZONE NUMBER OF THE ZONE HAVING THE LARGEST
C                       VALUE OF ARTIFICIAL VISCOSITY, Q.
C        JSMAX          ZONE NUMBER OF THE ZONE HAVING THE LARGEST
C                       STRESS.
C        JTS            ZONE NUMBER OF THE ZONE HAVING THE SMALLEST
C                       STABLE HYDRO TIME
C        PDTNEG         INTEGRATED NEGATIVE MOMENTUM FOR THE LAST
C                       JEDIT ZONE -- IT IS THE SUM OF ALL NEGATIVE
```

```
C                       (S+Q)*DTNH TERMS FOR THAT ZONE
C        PDTPOS         INTEGRATED POSITIVE MOMENTUM FOR THE LAST
C                       JEDIT ZONE
C        QMAX           THE MAXIMUM ARTIFICIAL VISCOSITY IN DYNES/CM**2
C                       IN THE MESH
C        SMAX           THE MAXIMUM STRESS IN THE MESH
C        X(JSMAX)       EULERIAN COORDINATE IN CENTIMETERS OF THE
C                       BOUNDARY OF THE ZONE WITH MAXIMUM STRESS
C        X(JQMAX)       EULERIAN COORDINATE IN CENTIMETERS OF THE
C                       BOUNDARY OF THE ZONE WITH MAXIMUM ARTIFICIAL
C                       VOSCOSITY
C
C                  *---*---*---*---*---*---*---*---*---*---*
C
C        SYMBOL DESIGNATIONS ON LOG PLOTS
C
C        W              NEGATIVE ABSCISSA
C        X              POSITIVE ABSCISSA
C        N              NEGATIVE STRESS
C        S              POSITIVE STRESS
C        Q              VISCOSITY
C        T              NEGATIVE PARTICLE VELOCITY
C        U              POSITIVE PARTICLE VELOCITY
C        E              SPECIFIC ENERGY
C        D              ZONE SIZE
C        H              TEMPERATURE
C        Z              ZONE IONIZATION
C        M              COMPRESSED VOLUME
C        L              EXPANDED VOLUME
C
C                  *---*---*---*---*---*---*---*---*---*---*
C
*CALL BLANK
*CALL AA
*CALL EQED
*CALL EQVP
*CALL PLOTCM
*CALL RZCOM
*CALL SPLLC
      DIMENSION A(120), EMVN(6), ZZCTRL(40)
      DIMENSION JP(5), JQMAX(6), JSSMAX(6), JTTMAX(6), KP(5), QMAX(6),
     1 QS(201), SQ(201), SSMAX(6), TTMAX(6)
C
      DATA BLANK, CON1, CON2, CON3, CON4, CON5, CON6, CON7, CON8, CON9,
     1 CON10, CON11, CON12, CON13, CON14, CON15, CON16 /' ','W','X','S',
     2 'N','Q','U','T','E','D','H','Z','M','L','-','B','+'/
C
      DO 10 I=1,5
      JP(I)=JBND(I)
      KP(I)=0
```

```
   10 CONTINUE
      JJJ=JFIN-1
      TT=TIME*1.E6
      DO 20 J=2,JFIN
      QS(J)=Q(J)*1.E-9
      SQ(J)=S(J)*1.E-9
   20 CONTINUE
      WRITE (1) N,TT,JJJ,NMTRLS,(X(J),SQ(J),QS(J),J=2,JFIN),JP,JJJ,KP
      IF (WTAPE) 30,60,30
   30 CONTINUE
      WRITE (6,1110)
      WRITE (6,1230) (DISCPT(I),I=1,8)
      WRITE (6,1210) N,TIME
      WRITE (6,1220)
      LINE=9
      J1=MAX0(JHAT,JSTAR)
      DO 40 J=1,J1
      WRITE (6,1180) J,CS(J),V(J),E(J),P(J),S(J),SD(J),U(J),ZM(J),TEMP(J
     1 ),ZFM(J),X(J),Q(J),QO(J),DV(J),EI(J),ITER(J),F(J),FO(J),EADD(J)
     2 ,TSPALL(J)
      LINE=LINE+4
      IF (LINE.LT.50) GO TO 40
      WRITE (6,1110)
      WRITE (6,1220)
      LINE=4
   40 CONTINUE
      JSTARD=MIN0(J1+1,JFIN)
      WRITE (6,1060)
      ZMAS=0.
      DO 50 J=2,JFIN
      ZMAS=ZMAS+ZM(J)
      WRITE (6,1070) PX(J),ZM(J),ZMAS,E(J)
   50 CONTINUE
   60 CONTINUE
      JSTARD=MIN0(JSTAR+1,JFIN)
C
C        TOTAL MOMENTUM CALCULATION
C
      EMVNEG=0.
      EMVPOS=0.
      DO 70 I=1,6
      EMVN(I)=0.
      JQMAX(I)=JBND(I)
      JSSMAX(I)=JBND(I)
      JTTMAX(I)=JBND(I)
      QMAX(I)=0.
      SSMAX(I)=0.
      TTMAX(I)=0.
      EMVN(I)=0.0
   70 CONTINUE
```

```
C
C     CALCULATE LAYER MOMENTA, TOTAL MOMENTUM IN MESH, AND ESTIMATE C
C     LATE TIME TRANSVERSE LINE LOAD (XNL)
C
      ESUM=0.
      EKSUM=0.
      XNL=0.
      M=1
      MS=1
      DO 190 J=2,JSTARD
      IF (J-1.EQ.JBND(M)) M=M+1
      IF (Q(J)-QMAX(M)) 90,90,80
   80 CONTINUE
      QMAX(M)=Q(J)
      JQMAX(M)=J
   90 CONTINUE
      DXX=X(J)-X(J-1)
      IF (TSPALL(J).NE.1.234) GO TO 100
      DXX=X(MS)-X(J-1)
      EMV=.5*ZM(J)*(US(MS)+U(J-1))
      MS=MS+1
      GO TO 110
  100 CONTINUE
      EMV=.5*ZM(J)*(U(J)+U(J-1))
  110 CONTINUE
      SL=1.5*SD(J)+6.0*GOVERK(J)*S(J)/(3.0+4.0*GOVERK(J))
      IF (TSPALL(J).EQ.8..OR.TSPALL(J).EQ.7.) SL=0.
      XNL=XNL+SL*DXX
      IF (EMV) 120,130,130
  120 EMVNEG=EMVNEG+EMV
      EMVN(M)=EMVN(M)+EMV
      GO TO 140
  130 EMVPOS=EMVPOS+EMV
      EMVN(M)=EMVN(M)+EMV
  140 CONTINUE
C
C     CALCULATE MAXIMUM STRESS AND MAXIMUM TENSION FOR EACH MATERIAL
C     LAYER
C
      IF (SSMAX(M)-S(J)) 150,160,160
  150 CONTINUE
      SSMAX(M)=S(J)
      JSSMAX(M)=J
  160 CONTINUE
      IF (TTMAX(M)+S(J)) 170,180,180
  170 CONTINUE
      TTMAX(M)=ABS(S(J))
      JTTMAX(M)=J
  180 CONTINUE
C
```

```
C          TOTAL ENERGY CALCULATION (IN CALORIES)
C
      ESUM=ESUM+(E(J)-7.5E10/XMW(M)*FLOAT(NATOM(M)))*ZM(J)/4.186E7
      EKSUM=EKSUM+ZM(J)*(U(J-1)**2+U(J-1)*U(J)+U(J)**2)/4.186E7/6.
  190 CONTINUE
      ETOTAL=ESUM+EKSUM
      JM=JSMAX+3
C
C          CALCULATE MOMENTUM OF MAIN PULSE
C
      EMVPL=0.
  200 EMVPL=EMVPL+U(JM)*.5*(ZM(JM+1)+ZM(JM))
      IF (JM-JSMAX) 210,220,220
  210 CONTINUE
      IF (JM-1) 230,230,220
  220 CONTINUE
      JM=JM-1
      IF (U(JM)) 230,230,200
  230 JM=JSMAX+4
C
C          CALCULATE MOMENTUM OF PRECURSOR
C
      EMVPR=0.
  240 EMVPR=EMVPR+U(JM)/2.*(ZM(JM)+ZM(JM+1))
      IF (JM-JSTAR) 250,250,260
  250 JM=JM+1
      GO TO 240
  260 EMVPP=EMVPL+EMVPR
      DTPP=EMVPP/SMAX
      DTPULS=EMVPL/SMAX
C
C          CALCULATE MAXIMUM POTENTIAL MOMENTUM OF VAPOR
C
      M=1
      EMVBM=0.
      EMVBM1=0.0
      EMVBM2=0.
      DO 280 J=2,JSTARD
      IF (J-1.EQ.JBND(M)) M=M+1
      DE=E(J)-7.5E10/XMW(M)*FLOAT(NATOM(M))-EQSTE(M)
      IF (DE) 290,290,270
  270 UAVG=(SQRT(U(J-1)*U(J-1)+2.*DE)+SQRT(U(J)*U(J)+2.*DE))/2.
      EMVBM=EMVBM+UAVG*ZM(J)
      UAVG=(U(J-1)+U(J))/2.0
      EMVBM1=EMVBM1+UAVG*ZM(J)
      UAVG1=-ABS(U(J-1))*U(J-1)+2.*DE
*IF DEF,B64
      X1=SIGN(1.0,UAVG1)
      UAVG2=-ABS(U(J))*U(J)+2.*DE
      X2=SIGN(1.0,UAVG2)
```

```
*ENDIF
*IF DEF,B32
      X1=DSIGN(1.D0,UAVG1)
      UAVG2=-ABS(U(J))*U(J)+2.*DE
      X2=DSIGN(1.D0,UAVG2)
*ENDIF
      UAVG=(X1*SQRT(X1*UAVG1)+X2*SQRT(X2*UAVG2))/2.0
  280 EMVBM2=EMVBM2+UAVG*ZM(J)
  290 CONTINUE
C
C     PUT TIME HISTORY OF MOMENTUM AND RERADIATION ON SEPARATE FILE
C     RATHER THAN IN DAYFILE.
C
      WRITE (9,1170) TIME,EMVBM,EMVBM1,EMVBM2,EMVNEG,EMVPOS,RERAD
C
C        PRINT OUTPUT VARIABLES
C
C     DTNR=0.9/DTRC
      JBND1=JBND(1)
      JBND2=JBND(2)
      JBND3=JBND(3)
      WRITE (6,1190)
      WRITE (6,1080) N,JFIN,JSTAR,JTS,TIME,DTNH,X(1),EMVNEG,EMVPOS,XNL
     1 ,ETOTAL
      WRITE (6,1090)
      DO 300 M=1,NMTRLS
      J1=JBND(M)
      J2=JSSMAX(M)
      J3=JTTMAX(M)
      J4=JQMAX(M)
      WRITE (6,1100) J1,J2,J3,J4,X(J1),X(J2),X(J3),X(J4),SSMAX(M),TTMAX
     1 (M),QMAX(M),EMVN(M)
  300 CONTINUE
      JS1=2
      JS2=JSTARD/3
      JS3=2*JS2
      JS4=JSTARD
      IF (ILOG) 630,310,630
  310 CONTINUE
      WRITE (3,1120) (DISCPT(I),I=1,8),JS1,JS2,JS3,JS4,S(JS1),S(JS2),S
     1 (JS3),S(JS4),U(JS1),U(JS2),U(JS3),U(JS4),E(JS1),E(JS2),E(JS3),E
     2 (JS4),V(JS1),V(JS2),V(JS3),V(JS4),X(JS1),X(JS2),X(JS3),X(JS4)
     3 ,TEMP(JS1),TEMP(JS2),TEMP(JS3),TEMP(JS4),ZFM(JS1),ZFM(JS2),ZFM
     4 (JS3),ZFM(JS4),(JBND(M),M=1,6)
C
C       PREPARE  LOG  PLOTS
C
      DO 320 J=1,120
      A(J)=BLANK
  320 CONTINUE
```

```
      JSSTAR=MINO(JSTARD+10,JFIN)
      M=1
      DO 620 J=2,JSSTAR
      IF (X(J)) 330,350,340
  330 CONTINUE
*IF DEF,B64
      NX=120.5+28.*(AMOD(AMOD(ALOG(ABS(X(J)))/2.3025851,4.)+4.,4.)-4.)
*ENDIF
*IF DEF,B32
      NX=120.5+28.*(DMOD(DMOD(DLOG(ABS(X(J)))/2.3025851DO,4.DO)+4.DO,4
     1 .DO)-4.DO)
*ENDIF
      A(NX)=CON1
      GO TO 350
  340 CONTINUE
*IF DEF,B64
      NX=20.5+100.*AMOD((2.0*ABS(X(J))),1.)
*ENDIF
*IF DEF,B32
      NX=20.5+100.*DMOD((2.DO*ABS(X(J))),1.DO)
*ENDIF
      A(NX)=CON2
  350 IF (S(J)) 360,390,360
  360 CONTINUE
*IF DEF,B64
      NX=120.5+28.*(AMOD(AMOD(ALOG(ABS(S(J))+.0001)/2.3025851,4.)+4.,4.)
     1 -4.)
*ENDIF
*IF DEF,B32
      NX=120.5+28.*(DMOD(DMOD(DLOG(ABS(S(J))+1.D-4)/2.3025851DO,4.DO)+4
     1 .DO,4.DO)-4.DO)
*ENDIF
      IF (S(J)) 380,370,370
  370 A(NX)=CON3
      GO TO 390
  380 A(NX)=CON4
  390 IF (Q(J).EQ.0.0) GO TO 400
*IF DEF,B64
      NX=120.5+28.*(AMOD(AMOD(ALOG(Q(J)+.0001)/2.3025851,4.)+4.,4.)-4.)
*ENDIF
*IF DEF,B32
      NX=120.5+28.*(DMOD(DMOD(DLOG(Q(J)+1.D-4)/2.3025851DO,4.DO)+4.DO.4
     1 .DO)-4.DO)
*ENDIF
      A(NX)=CON5
  400 IF (U(J)) 410,440,410
  410 CONTINUE
*IF DEF,B64
      NX=120.5+28.*(AMOD(AMOD(ALOG(ABS(U(J))+.0001)/2.3025851,4.)+4.,4.)
     1 -4.)
```

```
*ENDIF
*IF DEF,B32
      NX=120.5+28.*(DMOD(DMOD(DLOG(ABS(U(J))+1.D-4)/2.3025851D0,4.D0)+4
     1 .D0,4.D0)-4.D0)
*ENDIF
      IF (U(J)) 430,420,420
  420 A(NX)=CON6
      GO TO 440
  430 A(NX)=CON7
  440 IF (E(J)) 450,460,450
  450 CONTINUE
*IF DEF,B64
      NX=120.5+28.*(AMOD(AMOD(ALOG(ABS(E(J))+.0001)/2.3025851,4.)+4.,4.)
     1 -4.)
*ENDIF
*IF DEF,B32
      NX=120.5+28.*(DMOD(DMOD(DLOG(ABS(E(J))+1.D-4)/2.3025851D0,4.D0)+4
     1 .D0,4.D0)-4.D0)
*ENDIF
      A(NX)=CON8
  460 DXX=X(J)-X(J-1)
      IF (DXX) 480,480,470
  470 CONTINUE
*IF DEF,B64
      NX=120.5+28.*(AMOD(AMOD(ALOG(DXX)/2.3025851,4.)+4.,4.)-4.)
*ENDIF
*IF DEF,B32
      NX=120.5+28.*(DMOD(DMOD(DLOG(DXX)/2.3025851D0,4.D0)+4.D0,4.D0)-4
     1 .D0)
*ENDIF
      A(NX)=CON9
  480 IF (TEMP(J)) 500,500,490
  490 CONTINUE
*IF DEF,B64
      NX=120.5+28.0*(AMOD(AMOD(ALOG(TEMP(J)+.0001)/2.3025851,4.)+4.,4.)-
     1 4.)
*ENDIF
*IF DEF,B32
      NX=120.5+28.0*(DMOD(DMOD(DLOG(TEMP(J)+1.D-4)/2.3025851D0,4.D0)+4
     1 .D0,4.D0)-4.D0)
*ENDIF
      A(NX)=CON10
  500 CONTINUE
*IF DEF,B64
      NX=8.5+AMOD(ZFM(J),112.0)
*ENDIF
*IF DEF,B32
      NX=8.5+DMOD(ZFM(J),112.D0)
*ENDIF
      A(NX)=CON11
      IF (V(J)-1.) 510,510,520
  510 CONTINUE
*IF DEF,B64
      NX=120.5+28.*(AMOD(AMOD(ALOG(.01*(1./V(J)-.99))/2.3025851,4.)+4.,4
     1 .)-4.)
*ENDIF
*IF DEF,B324
      NX=120.5+28.*(DMOD(DMOD(DLOG(1.D-2*(1.D0/V(J)-9.9D-1))/2.3025851D0
     1 ,4.D0)+4.D0,4.D0)-4.D0)
*ENDIF
      A(NX)=CON12
      GO TO 530
  520 CONTINUE
*IF DEF,B64
      NX=120.5+28.*(AMOD(AMOD(ALOG(1./V(J))/2.3025851,4.)+4.,4.)-4.)
*ENDIF
*IF DEF,B32
      NX=120.5+28.*(DMOD(DMOD(DLOG(1.D0/V(J))/2.3025851D0,4.D0)+4.D0,4
     1 .D0)-4.D0)
*ENDIF
      A(NX)=CON13
  530 IF (JBND(M)-J) 570,540,570
  540 M=M+1
      DO 560 I=8,120
      IF (A(I)-BLANK) 560,550,560
  550 A(I)=CON14
  560 CONTINUE
      A(15)=CON15
      A(105)=CON15
  570 IF (MOD(J,10)) 600,580,600
  580 DO 590 I=8,120,28
      A(I)=CON16
  590 CONTINUE
      WRITE (3,1130) J,(A(I),I=8,120)
      GO TO 610
  600 WRITE (3,1140) (A(I),I=1,120)
  610 DO 620 I=1,120
  620 A(I)=BLANK
C
C                END PREPARE LOG PLOTS
C
C     PREPARE LINEAR PLOTS OF S VS X
C
  630 IF (ILIN) 1050,640,1050
  640 MX=0.
      MP=0.
      KPLUS=0
      XINC=100.
      M=1
      JJ=JBND(1)
      IF (JJ) 650,650,660
  650 JJ=JFIN
  660 JM=MINO(JSTAR+2,JFIN+1)
  670 JM=JM-1
      IF (ABS(S(JM))-.01*ABS(SMAX)) 670,670,680
  680 JM=MINO(JM+1,JFIN)
      JT=1
  690 JT=JT+1
      IF (ABS(S(JT))-.01*ABS(SMAX)) 690,690,700
  700 JT=JT-1
      JT=MAXO(JT,2)
  710 MP=MP+1
      IF (ABS(SMAX)-SCALE(MP)) 720,720,710
  720 PWX=X(JM)-X(JT)
  730 MX=MX+1
      IF (PWX-SCAX(MX)) 740,740,730
  740 DSCAX=SCAX(MX)/XINC
*IF DEF,B64
      XPES=FLOAT(IFIX(X(JT)/DSCAX))*DSCAX
*ENDIF
*IF DEF,B32
      XPES=FLOAT(IDINT(X(JT)/DSCAX))*DSCAX
*ENDIF
      SCX=SCAX(MX)
      SCY=SCALE(MP)
      WRITE (3,1160) (DISCPT(I),I=1,8),N,TIME,SCALE(MP),XPES
      GO TO 760
  750 JT=JT+1
      IF (JT.GE.JFIN) GO TO 1050
  760 XAVG=(X(JT+1)+X(JT))/2.
      IF (XAVG-X(JFIN)) 770,1050,1050
  770 IF (XAVG-XPES) 750,780,780
  780 DELX=(X(JT+1)-X(JT-1))/2.
      IF (JT.EQ.1) DELX=(X(2)-X(1))/2.
      IF (DELX) 790,790,800
  790 SC=0.
      GO TO 810
  800 D1=(DELX-XAVG+XPES)/DELX
      SC=S(JT)+(S(JT+1)-S(JT))*D1
  810 IF (Q(JT)+Q(JT+1)) 830,830,820
  820 QQ=SC+Q(JT)+(Q(JT+1)-Q(JT))*D1
*IF DEF,B64
      NX=60.5+AMOD(50.*QQ/SCALE(MP),50.)
*ENDIF
*IF DEF,B32
      NX=60.5+DMOD(5.D0*QQ/SCALE(MP),5.D0)
*ENDIF
      A(NX)=CON5
  830 NX=60.5+50.*SC/SCALE(MP)
      NX=MAXO(NX,1)
      NX=MINO(NX,120)
      A(NX)=CON3
      IF (A(60)-BLANK) 850,840,850
  840 A(60)=CON16
  850 KSW=1
      IF (KPLUS-5) 890,860,890
  860 KPLUS=0
      KSW=2
      DO 880 I=10,110,10
      IF (A(I)-BLANK) 880,870,880
  870 A(I)=CON16
  880 CONTINUE
  890 KPLUS=KPLUS+1
      XPES=XPES+DSCAX
  900 IF (XPES-X(JJ)) 1000,950,910
  910 IF (XPES-DSCAX-X(JJ)) 950,950,920
  920 M=M+1
      M=MINO(M,NMTRLS)
      JJ=JBND(M)
      IF (JJ) 940,930,940
  930 JJ=JFIN
  940 GO TO 900
  950 M=M+1
      M=MINO(M,NMTRLS)
      DO 970 I=10,110
      IF (A(I)-BLANK) 970,960,970
  960 A(I)=CON14
      A(15)=CON15
      A(105)=CON15
  970 CONTINUE
      JJ=JBND(M)
      IF (JJ) 990,980,990
  980 JJ=JFIN
  990 CONTINUE
 1000 GO TO (1020,1010), KSW
      CALL GOTOER
 1010 WRITE (3,1150) XPES,(A(I),I=10,110)
      GO TO 1030
 1020 WRITE (3,1140) (A(I),I=1,110)
 1030 DO 1040 I=1,110
 1040 A(I)=BLANK
      IF (XPES-X(JM)) 760,1050,1050
 1050 WRITE (3,1200)
      RETURN
C
 1060 FORMAT (1H0,8X,2HPX,8X,2HZM,6X,4HZMAS,9X,1HE/)
 1070 FORMAT (5X,1P4E10.3)
 1080 FORMAT (//,5X,5HCYCLE,2X,4HJFIN,1X,5HJSTAR,3X,3HJTS,6X,4HTIME,6X,4
     1 HDTNH,6X,4HX(1),4X,6HEMVNEG,4X,6HEMVPOS,1X,9HLINE LOAD,4X,6HETOTA
     2L/5X,3(I5,1X),I5,1P7E10.3)
```

```
1090 FORMAT (/6X,4HJBND,1X,5HJSMAX,1X,5HJTMAX,1X,5HJQMAX,3X,7HX(JBND),2
    1 X,8HX(JSMAX),2X,8HX(JTMAX),2X,8HX(JQMAX),6X,4HSMAX,6X,4HTMAX,6X,4
    2 HQMAX,2X,8HMOMENTUM)
1100 FORMAT (4X,4(1X,I5),1P8E10.3)
1110 FORMAT (1H1)
1120 FORMAT (/1X,8A10/5H ZONE,7X,4(I3,9X)/4H S   ,4E12.3/4H U   ,4E12.3/4
    1 H E   ,4E12.3/4H V   ,4E12.3/4H X   ,4E12.3/5H TEMP,4E12.3/4H ZF  ,4E
    2 12.3//15H BOUNDARIES AT ,6I4//52H LINEAR X PLOT, FULL SCALE=0.5 C
    3M (100 PRINT WHEELS)/1X,31HLINEAR ZF PLOT, FULLSCALE = 112//1H0,4H
    4ZONE,2X,4H10-4,23X,4H10-3,24X,4H10-2,24X,4H10-1,25X,1H1/1H ,6X,1H+
    5 ,27X,1H+,27X,1H+,27X,1H+,27X,1H+)
1130 FORMAT (1H ,I4,2X,113A1)
1140 FORMAT (120A1)
1150 FORMAT (1H ,F7.4,1X,101A1)
1160 FORMAT (27H1PLOT OF LINEAR STRESS VS X/1H ,8A10,11H CYCLE NO.=,I4,
    1 10X,5HTIME=,E10.3/14H TOP OF PAGE =,E12.3,24H DYNES/SQUARE CENTIM
    2ETER//1H0,6X,4H-2.5,6X,4H-2.0,6X,4H-1.5,6X,4H-1.0,6X,4H-0.5,6X,4H
    30.0,7X,3H0.5,7X,3H1.0,7X,3H1.5,7X,3H2.0,7X,3H2.5/1H0,6X,4H-5.0,6X,
    4 4H-4.0,6X,4H-3.0,6X,4H-2.0,6X,4H-1.0,7X,3H0.0,7X,3H1.0,7X,3H2.0,7
    5 X,3H3.0,7X,3H4.0,7X,3H5.0/1H0,5X,5H-10.0,6X,4H-8.0,6X,4H-6.0,6X,4
    6 H-4.0,6X,4H-2.0,7X,3H0.0,7X,3H2.0,7X,3H4.0,7X,3H6.0,7X,3H8.0,6X,4
    7 H10.0/1H0,F7.4,1X,1H+,10(9X,1H+))
1170 FORMAT (1X,1P7E12.3)
1180 FORMAT (//5X,I5,1P10E12.4/10X,1P5E12.4,I12,1P4E12.4)
1190 FORMAT (1HQ)
1200 FORMAT (1HR)
1210 FORMAT (20H  TIME EDIT AT CYCLE,I5,5X,5H TIME,1PE15.5,4H SEC//)
1220 FORMAT (//6X,2H J,6X,3H CS,10X,2H V,10X,2H E,10X,2H P,10X,2H S,9X,
    1 3H SD,10X,2H U,9X,3H ZM,8X,5H TEMP,8X,4H ZFM/15X,2H X,10X,2H Q,9X
    2 ,3H QO,9X,3H DV,9X,3H EI,8X,5H ITER,9X,2H F,9X,3H FO,8X,5H EADD,6
    3 X,7H TSPALL)
1230 FORMAT (1H0,T26,8A10)
     END
```

```
*DECK EQST
      SUBROUTINE EQST
*IF DEF,B32
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
*ENDIF
C
*CALL BLANK
*CALL AE
*CALL EQED
*CALL EQFL
*CALL EQVP
*CALL HYEQ
*CALL INDX
*CALL PLOTCM
*CALL RZCOM
*CALL SPLLC
C     THIS SUBROUTINE CALCULATES NEW TEMPERATURES IN THE MESHES BY
C             MAKING ITERATIVE GUESSES.
C
      M=1
      NCOUNT=0
      J1=MAX0(JSTAR,JHAT)
C
C     MAKE FIRST GUESS AT NEW TEMPERATURES IN THE MESHES
C
      FO(1)=F(1)
      DO 200 J=2,JFIN
      IF (J-1.EQ.JBND(M)) M=M+1
      ZFM(J)=ZFM(J)/FLOAT(NATOM(M))
      FO(J)=F(J)
      TKEEP(J)=TEMP(J)
      IF (IFLOW.NE.0) F(J)=0.0
      P(J)=P(J)+Q(J)+QO(J)
      ITER(J)=4
      IF (IDIF.EQ.0) ITER(J)=1
      IF (ICON.EQ.0) ITER(J)=2
      IF (ICON.EQ.0.AND.IDIF.EQ.0) ITER(J)=3
      IF (TIME.GT.SSTOPM) GO TO 190
      DEDTP=1.25E8/XMW(M)*FLOAT(NATOM(M))
      DPDTP=8.31E7/XMW(M)/V(J)*FLOAT(NATOM(M))
      DELRO=1./V(J)-1./(V(J)-DV(J))
      ZSTAR=0.0
      IF (ZFM(J).GT.10.0) ZSTAR=ZFM(J)
      NEM=NELEM(M)
      XNATOM=6.02E23/XMW(M)/V(J)*FLOAT(NATOM(M))
      XN1=2.43E15/XNATOM
      ENU=1./V(J)/RHO(M)
      IF (ENU.LT.1.) GO TO 10
C
```

```
C     MATERIAL IS COMPRESSED
C
      W1=2.0
      W2=3.*EQSTG(M)
      DES1=0.0
      GO TO 90
C
C     MATERIAL IS EXPANDED
C
   10 ALF=EQSTH(M)+(EQSTG(M)-EQSTH(M))*SQRT(ENU)
      IF (ABS(ALF-EQSTH(M)).LE.1.E-3) GO TO 20
      GO TO 30
   20 W1=1.
      GO TO 40
   30 W1=2.*(EQSTG(M)-ALF)/(ALF-EQSTH(M))
      W1=(2.+W1)/(1.+W1)
   40 W2=1.5*ALF*W1
      V1=(V(J)-DV(J))*RHO(M)
      ENU2=EQSTN(M)*(1.-V1)*V1
      IF (ENU2.LT.-10.0) GO TO 50
      ES1O=EQSTE(M)*(1.-EXP(ENU2))
      GO TO 60
   50 ES1O=EQSTE(M)
   60 V1=V(J)*RHO(M)
      ENU2=EQSTN(M)*(1.-V1)*V1
      IF (ENU2.LT.-10.0) GO TO 70
      ES1N=EQSTE(M)*(1.-EXP(ENU2))
      GO TO 80
   70 ES1N=EQSTE(M)
   80 DES1=ES1O-ES1N
C
C     FIRST GUESS AT NEW TEMPERATURES
C
   90 TEMPJ=TEMP(J)
      IF (ZFM(J).GT.10.0) GO TO 110
      ITRY=1
  100 ZSTAR=ZSTAR+1
  110 DEDT=DEDTP*(W1+3.*ZSTAR)
      DPDT=DPDTP*(W2+2.*ZSTAR)
      TEMPJ=(TEMP(J)*DEDT+EADD(J)+DES1-(Q(J)+QO(J))/2.*DV(J))/(DEDT-V(J)
     1 **2*DPDT*DELRO)
      TEMPJ=TEMPJ+(FO(J)-FO(J-1))*DTNH/ZM(J)/(DEDT-V(J)**2*DPDT*DELRO)
      IF (TEMPJ.LT.300.0) TEMPJ=300.0
      IF (TEMP(J).LT.1.2E4) GO TO 180
      IF (ZFM(J).GT.10.0) GO TO 180
      IF (ITRY.EQ.2) GO TO 180
      XTEV=TEMPJ/1.16E4
      FXPC=XN1*TEMPJ**1.5
      ZSTAR1=0.0
      DO 170 N1=1,NEM
      N2=IELEM(M,N1)
      NT=NTBL(N2)
      DO 120 K=1,NT
C     AVOID EXP UNDERFLOW
      ARGEXP=-XI(N2,K)/XTEV
      EXPR=0.0
      IF (ARGEXP.GT.-675.) EXPR=EXP(ARGEXP)
      XP=FXPC*EXPR
      IF (XP.LT.FLOAT(K)) GO TO 130
  120 CONTINUE
  130 K=K-1
      DO 160 K1=1,10
      IF (K.EQ.0) GO TO 140
      XITMP=XI(N2,K)+FLOAT(K1)/10.0*(XI(N2,K+1)-XI(N2,K))
      GO TO 150
  140 XITMP=FLOAT(K1)/10.0*XI(N2,1)
  150 XP=FXPC*EXP(-XITMP/XTEV)
      IF (XP.LT.FLOAT(K)+FLOAT(K1)/10.0) GO TO 170
  160 CONTINUE
  170 ZSTAR1=ZSTAR1+AF(M,N2)*(FLOAT(K)+FLOAT(K1-1)/10.0)
      IF (ZSTAR1.GT.ZSTAR) GO TO 100
      ZSTAR=(ZSTAR+ZSTAR1)/2.0
      ITRY=2
      GO TO 110
  180 TEMP(J)=TEMPJ
  190 CONTINUE
*IF DEF,B64
      TEMP(J)=AMAX1(TEMP(J),300.0)
*ENDIF
*IF DEF,B32
      TEMP(J)=DMAX1(TEMP(J),3.D2)
*ENDIF
      TEMPO(J)=TEMP(J)
      ZFM(J)=ZFM(J)*FLOAT(NATOM(M))
      IF (TEMP(J).GT.1.2E4.AND.J.GT.JHAT) JHAT=J
      IF (TEMP(J).GE.301.0.AND.J.GT.JSTAR) JSTAR=J
      IF (J.LE.J1) GO TO 200
      IF (TEMP(J).LT.301.0) ITER(J)=0
      IF (TEMP(J).LT.301.0) GO TO 210
  200 CONTINUE
  210 KPRIN=0
      IF ((JTRY.EQ.2).OR.(JPRIN.EQ.1.AND.N/NPRIN*NPRIN.EQ.N)) KPRIN=1
C
C     FIND NEW IONIZATION IN THE MESHES AND NEW ENERGY FLUXES AT THE
C             BOUNDARIES IF ITER(J) NOT EQUAL TO ZERO.
C
      ICHCK=1
  220 CALL FLOION
C
C     NOW DETERMINE AMOUNT BY WHICH TEMPERATURE GUESSES ARE INCORRECT.
```

```
C
      JCOUNT=1
      J1=MAX0(JSTAR,JHAT)
      CALL PT
      NCOUNT=NCOUNT+1
C
      IF (NCOUNT.LT.100+J1.OR.JCOUNT.EQ.J1) GO TO 290
      WRITE (6,550) NCOUNT,N,DTNH
      DO 230 J=2,J1
      IF (ITER(J).NE.0) WRITE (6,560) J
  230 CONTINUE
      GO TO (240,280), JTRY
      CALL GOTOER
  240 F(1)=FO(1)
      X(1)=X(1)-U(1)*DTNH
      M=1
      MS=1
      DO 270 J=2,J1
      TEMP(J)=TKEEP(J)
      X(J)=X(J)-U(J)*DTNH
      IF (TSPALL(J).NE.1.234) GO TO 250
      XS(MS)=XS(MS)-US(MS)*DTNH
      MS=MS+1
  250 CONTINUE
      V(J)=V(J)-DV(J)
      DV(J)=0.0
      P(J)=P(J)-Q(J)-QO(J)
      S(J)=P(J)-SD(J)
      Q(J)=QO(J)
C
C     SET SPALL FLAGS FOR ACTIVE ZONES FOR JTRY = 1.  USE SAME RULES AS
C     FOR JTRY = 2
C
      IF (J.GT.JBND(M)) M=M+1
      IF (TSPALL(J).EQ.1.234) GO TO 270
      IF (TSPALL(J).EQ.8.) GO TO 270
      IF (TSPALL(J).EQ.7.) GO TO 260
      IF (E(J).LT.EM(M)) GO TO 270
      TSPALL(J)=7.
  260 CONTINUE
      IF (E(J).LT.EQSTE(M)) GO TO 270
      TSPALL(J)=8.
  270 F(J)=FO(J)
      DTN=DTN-0.2*DTNH
      DTNH=0.8*DTNH
      JTRY=2
      RETURN
  280 KCOUNT=KCOUNT+1
      IF (KCOUNT.LT.10) GO TO 310
      WRITE (6,540)
      CALL EDIT
      STOP 1
C
C     CHECK TO SEE IF ALL MESHES HAVE CONVERGED TO A SATISFACTORY
C             NEW TEMPERATURE.  IF NOT, CALL FLOION.
C
  290 IF (JCOUNT.EQ.J1) GO TO 310
      DO 300 J=2,J1
      IF (ITER(J).EQ.0.OR.ITER(J+1).EQ.0) GO TO 300
      IF (ABS(TEMPO(J+1)-TEMPO(J)).LT.1.E2) GO TO 300
      IF (TEMPO(J).GT.TEMPO(J+1).AND.TEMP(J).GT.TEMP(J+1)) GO TO 300
      IF (TEMPO(J).LT.TEMPO(J+1).AND.TEMP(J).LT.TEMP(J+1)) GO TO 300
      TEMP(J)=(TEMP(J)*ZM(J)+TEMP(J+1)*ZM(J+1))/(ZM(J)+ZM(J+1))
      TEMP(J+1)=TEMP(J)
  300 CONTINUE
      IF (JCOUNT.LT.J1) GO TO 220
  310 JTRY=1
      IF (J1.EQ.JFIN) GO TO 330
C
C     ADD ENERGY TO ZONES WHICH ARE NOT YET ACTIVE
C
      J2=J1+1
      DO 320 J=J2,JFIN
      IF (J-1.EQ.JBND(M)) M=M+1
      E(J)=E(J)+EADD(J)+(F(J)-F(J-1))/ZM(J)*DTNH
      TEMP(J)=E(J)*XMW(M)/FLOAT(NATOM(M))/2.50E8
      IF (EADD(J).EQ.0.0) TEMP(J)=300.0
      EADD(J)=0.0
      P(J)=2.493E8*EQSTG(M)/XMW(M)/V(J)*(TEMP(J)-300.)
      IF (P(J).LT.0.0) P(J)=0.0
      S(J)=P(J)-SD(J)
*IF DEF,B64
      PSMAX(J)=AMAX1(PSMAX(J),S(J))
      PSMIN(J)=AMIN1(PSMIN(J),S(J))
*ENDIF
*IF DEF,B32
      PSMAX(J)=DMAX1(PSMAX(J),S(J))
      PSMIN(J)=DMIN1(PSMIN(J),S(J))
*ENDIF
  320 CONTINUE
C
C     CALL FLOION ONE MORE TIME TO CALCULATE ENERGY DEPOSITION RATES IN
C     THE ZONES FOR THE NEXT CYCLE
C
  330 ICHCK=0
      IF (TIME.GT.SSTOPM) GO TO 340
      IF (N/NDEP*NDEP.EQ.N) CALL FLOION
  340 ICHCK=1
C
C     STORE VALUES OF NEWLY FOUND VARIABLES, FIND SMAX AND JSMAX,
C               CALCULATE SHOCK SPEEDS IN THE MESHES, FIND ZONE WHICH
C               CONTROLS NEXT TIME STEP AND SK2M IN THAT ZONE.
C     FIND LARGEST STABLE ENERGY TRANSFER TIME, DTRC, AND ZONE NUMBER
C     IN WHICH THIS OCCURS,  ALSO FIND SMALLEST TIME IN WHICH SOME ZONE
C     ENERGY WILL BE ALTERED BY 20 PERCENT DUE TO ENERGY TRANSFER ONLY
C     THIS TIME IS CALLED DTPRIM
C
      DTRC=1.0
      DTRC1=0.0
      JTR=0
      SMAX=0.0
      SK2M=0.0
      DTPRIM=10
      M=1
      MS=1
      DO 530 J=2,J1
      IF (J-1.EQ.JBND(M)) M=M+1
      ZFM(J)=ZFM(J)/FLOAT(NATOM(M))
      IF (ITER(J).NE.0) TEMP(J)=TKEEP(J)
      IF (ITER(J).NE.0) GO TO 350
      P(J)=PN(J)
      E(J)=ET(J)
  350 S(J)=P(J)-SD(J)
C
C     SET SPALL FLAGS FOR MELT AND VAPORIZATION UNLESS MATERIAL
C     HAS ALREADY BEEN SPALLED.
C
      IF (TSPALL(J).EQ.1.234) GO TO 370
      IF (TSPALL(J).EQ.7.) GO TO 360
      IF (TSPALL(J).EQ.8.) GO TO 370
      IF (E(J).LT.EM(M)) GO TO 370
      TSPALL(J)=7.
  360 CONTINUE
      IF (E(J).LT.EQSTE(M)) GO TO 370
      TSPALL(J)=8.
  370 CONTINUE
      IF (TSPALL(J).EQ.1.234) MS=MS+1
*IF DEF,B64
      PSMAX(J)=AMAX1(PSMAX(J),S(J))
      PSMIN(J)=AMIN1(PSMIN(J),S(J))
*ENDIF
*IF DEF,B32
      PSMAX(J)=DMAX1(PSMAX(J),S(J))
      PSMIN(J)=DMIN1(PSMIN(J),S(J))
*ENDIF
      IF (ABS(S(J)).LT.ABS(SMAX)) GO TO 380
      SMAX=S(J)
      JSMAX=J
  380 V1=RHO(M)*V(J)
      ENU=1./V1
      EMU=1./V1-1.
      IF (EMU.LT.0.) GO TO 410
C
C     MATERIAL IS COMPRESSED.
C
      DXZ=0.3325*(3.*EQSTG(M)+ZFM(J))/(2.+ZFM(J))/RHO(M)
      XZ=1.-DXZ*RHO(M)*(1.-ENU)
      XZ2=XZ**2
      DXY2=2.*DXZ*RHO(M)*(E(J)-EI(J))
      XY2=DXY2/V(J)
      IF (CUSPA(M).LE.0.) GO TO 390
      ARG=EMU-CUSPA(M)
      IF (ARG.LE.0.) GO TO 390
      XY1=CUSP1(M)+CUSPC(M)*ARG+CUSPD(M)*ARG**2+CUSPS(M)*ARG**3
      DXY1=CUSPC(M)+2.*CUSPD(M)*ARG+3.*CUSPS(M)*ARG**2
      GO TO 400
  390 XY1=((EQSTS(M)*EMU+EQSTD(M))*EMU+EQSTC(M))*EMU
      DXY1=(3.*EQSTS(M)*EMU+2.*EQSTD(M))*EMU+EQSTC(M)
  400 DPDRO=((XY1+XY2)*DXZ-XZ*(DXY1/RHO(M)+DXY2))/XZ2
      DPDRO=-1.*DPDRO
      DEDT=2.50E8/XMW(M)*(1.+ZFM(J)/2.)*FLOAT(NATOM(M))
      GO TO 440
C
  410 ALF=EQSTH(M)+(EQSTG(M)-EQSTH(M))*SQRT(ENU)
      IF (ABS(ALF-EQSTH(M)).GT.1.E-3) GO TO 420
      W1=1.0
      GO TO 430
  420 W1=2.*(EQSTG(M)-ALF)/(ALF-EQSTH(M))
      W1=(2.+W1)/(1.+W1)
  430 DPDRO=P(J)*V(J)*(1.0+0.665*(1.5*ALF*W1+ZFM(J))/(W1+ZFM(J)))
      DEDT=1.25E8/XMW(M)*(W1+ZFM(J))*FLOAT(NATOM(M))
C
C     CALCULATE SHOCK SPEEDS IN MESHES AND THE QUANTITY (1. / SMALLEST
C              REQUIRED HYDRO TIME STEP).
C
  440 IF (DPDRO.GT.0.) GO TO 450
      V1=1./ENU
      ENU2=EQSTN(M)*(1.-V1)*V1
      EXPR=0.
      IF (ENU2.GT.-675.) EXPR=EXP(ENU2)
      ES1=EQSTE(M)*(1.-EXPR)
      ES10=FLOAT(NATOM(M))/XMW(M)
      TREF=ES1/(2.5E8*ES10*(1.+.5*ZFM(J)))
      DFDT=(TEMP(J)-TREF)*ES10
      IF (DFDT.EQ.0.) DFDT=-1.
      DPDT=EQSTE(M)/(2.5E8*(1.+.5*ZFM(J)))
      DPDT=DPDT*EXPR*EQSTN(M)
      DPDT=DPDT*(2.*V1**3-V1*V1)/RHO(M)
      DPDRO=DPDRO+P(J)*DPDT/DFDT
      IF (DPDRO.LE.0.) GO TO 460
  450 CS(J)=SQRT(DPDRO)
  460 CONTINUE
      DXZ=X(J)-X(J-1)
      IF (TSPALL(J).EQ.1.234) DXZ=XS(MS)-X(J-1)
      IF (MFLAG(M).GT.0.AND.J.GT.JB(M)) GO TO 470
C
C     CALCULATION OF MAXIMUM TIME STEP FOR NONDISPERSIVE MATERIAL
C
      SK2M1=CS(J)/DXZ
      IF (Q(J).GT.0.) SK2M1=SK2M1+(2.*C1*CS(J)-4.*CO*CO*DV(J)*ZM(J)/DTNH
     1 )/DXZ
      GO TO 480
  470 CONTINUE
C
C     CALCULATION OF MAXIMUM TIME STEP FOR DISPERSIVE MATERIAL
C
      DU=U(J)-U(J-1)
      IF (TSPALL(J).EQ.1.234) DU=US(MS)-U(J-1)
*IF DEF,B64
      XX=2.*EQSTA(M)*(RHO(M)*CH(M))**2/(OMEGA(M)*CS(J)*ZM(J)/V(J))+2.*CO
     1 *CO*(ABS(AMIN1(DU,0.0)))/CS(J)
*ENDIF
*IF DEF,B32
      XX=2.*EQSTA(M)*(RHO(M)*CH(M))**2/(OMEGA(M)*CS(J)*ZM(J)/V(J))+2.*CO
     1 *CO*(ABS(DMIN1(DU,0.DO)))/CS(J)
*ENDIF
      XXX=1.+VAMU(J)/(CS(J)**2/V(J))
      SK2M1=(CS(J)/DXZ)*(XXX/(-XX+SQRT(XX*XX+XXX*(1.+8.*(RHO(M)*CH(M)/
     1 (OMEGA(M)*ZM(J)))**2))))
  480 IF (SK2M-SK2M1) 490,490,500
  490 SK2M=SK2M1
      JTS=J
  500 CONTINUE
C
C     REDUCE ALLOWABLE TIME STEP IF NUMBER OF ITERATIONS ON TEMPERATURES
C     IS GREATER THAN ( 40+J1)
C
      IF (NCOUNT.LT.40+J1) GO TO 510
*IF DEF,B64
      SK2M=AMAX1(SK2M,0.81/DTNH)
*ENDIF
*IF DEF,B32
      SK2M=DMAX1(SK2M,0.81/DTNH)
*ENDIF
      JTS=0
  510 CONTINUE
C
C     CALCULATE THE MINIMUM TIME OVER ALL MESHES IN WHICH THE ENERGY
C             WILL HAVE BEEN ALTERED BY   ONE  PERCENT DUE TO ENERGY
C             TRANSFER ALONE.
```

```
C
      IF (IDIF.NE.0.AND.ICON.NE.0) GO TO 520
      IF (F(J).EQ.F(J-1)) GO TO 520
      DTPR1=ABS(.01*E(J)*ZM(J)/(F(J)-F(J-1)))
      IF (DTPR1.GT.DTPRIM) GO TO 520
      DTPRIM=DTPR1
      JTR=J
  520 ZFM(J)=ZFM(J)*FLOAT(NATOM(M))
      EADD(J)=0.0
  530 CONTINUE
C
C     CALCULATE THE TOTAL ENERGY RERADIATED FROM THE FRONT AND REAR
C     SURFACES OF THE MATERIAL UP TO THE CURRENT TIME
C
      RERAD=RERAD+(F(1)-F(JFIN))/4.186E7*DTNH
      RETURN
C
  540 FORMAT (83H0***** PROBLEM SHUT-DOWN BECAUSE OF REPEATED NON-CONVER
     1GENCE OF TEMPERATURES ***** )
  550 FORMAT (57H ITERATION FOR NEW TEMPERATURES DISCONTINUED AT NCOUNT
     1= ,I10,3X,9H CYCLE = ,I10,3X,6HDTNH =,E10.3)
  560 FORMAT (10H J EQUALS ,I10,5X,17HHAS NOT CONVERGED)
      END
```

```
*DECK FINDRZ
      SUBROUTINE FINDRZ (DX,DT,N,RZ1)
*IF DEF,B32
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
*ENDIF
C
C     FINDRZ CALCULATES THE COMMON RATIO OF A GEOMETRIC PROGRESSION OF
C     FINDRZ CALCULATES THE COMMON RATIO OF A GEOMETRIC PROGRESSION OF
C     N TERMS, WITH AN INITIAL CELL SIZE OF DX AND A TOTAL SIZE OF DT.
C
      S=1.
      RZ1=0.
      DR=1.
   10 CONTINUE
      RZ1=RZ1+S*DR
      IF (RZ1.EQ.1.) GO TO 20
      D1=DX*(RZ1**N-1.)/(RZ1-1.)
      GO TO 30
   20 CONTINUE
      D1=N*DX
   30 CONTINUE
      IF (S.GT.0..AND.D1.LT.DT) GO TO 10
      IF (S.LT.0..AND.D1.GT.DT) GO TO 10
      S=-S
      DR=.5*DR
      D2=(DT-D1)/DT
      IF (ABS(D2).GT.1.E-6) GO TO 10
      RETURN
C
C**  THIS PROGRAM VALID ON FTN4 AND FTN5 **
      END
```

```
*DECK FLOION
      SUBROUTINE FLOION
*IF DEF,B32
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
*ENDIF
C
      REAL LGDEL
C
*CALL BLANK
*CALL AA
*CALL AB
*CALL AC
*CALL EQFL
*CALL INDX
C     THIS ROUTINE CALCULATES THE DEGREE OF IONIZATION IN THE MESHES
C             AND THE ENERGY FLOW ACROSS MESH BOUNDARIES WHEN GIVEN
C             MESH VOLUMES AND TEMPERATURES.
C
      ICHCK1=ICHCK+1
      M=1
      NKEEP=N
      DO 10 J=2,JFIN
   10 XFX(J)=0.0
C
C     CALCULATE THE IONIZATION OF EACH ELEMENT OCCURING IN EACH MESH
C
      DO 130 J=2,JFIN
      IF (J-1.EQ.JBND(M)) M=M+1
      NEM=NELEM(M)
      IF (J-JHAT) 30,30,20
   20 CONTINUE
      GO TO (120,130), ICHCK1
      CALL GOTOER
   30 IF (ITER(J).EQ.0.AND.ICHCK.EQ.1) GO TO 130
      IF (TEMP(J)-1.2E4) 40,40,50
   40 IF (EI(J).GT.1.E-20) GO TO 50
      GO TO (120,130), ICHCK1
      CALL GOTOER
   50 CALL SAHA
C
C     IONIZATION LEVELS OF ALL ELEMENTS ARE CHOSEN PRCPERLY - THE
C     ENERGY USED IN CAUSING IONIZATION IN MESH J IS CALCULATED AND
C     THE IONIZATION LEVEL IN EACH ELEMENT AND THE LEVEL FOR THE
C     MATERIAL ARE STO..ED
C
      EI(J)=0.0
      DO 70 N1=1,NEM
      N=IELEM(M,N1)
      KN=KMAX(N)
      DO 60 K=1,KN
      L=NI(K.N)
      IF (L.LE.0) GO TO 60
      EI(J)=EI(J)+R(K,N)*EN(N,L)*AF(M,N)*FLOAT(NATOM(M))*9.632E11/XMW(M)
      IF (J.NE.JTS) GO TO 60
      IF (KPRIN.EQ.1) WRITE (6,190) MATL(M),NAMEL(N),KN,L,J,EI(J),R(K,N)
     1 ,EN(N,L),AF(M,N)
   60 CONTINUE
   70 ZF(N,J)=ZF1(N)
      ZFM(J)=ZSTAR
      IF (MCOUNT.GT.500) GO TO 80
      IF (ABS(ZSTAR-ZSTAR1).GT.0.01*ZSTAR1.AND.ZSTAR.GE.1.E-40) GO TO 80
C
      IF (ICHCK.EQ.0) GO TO 110
      IF (KPRIN.EQ.0) GO TO 130
      IF (J.NE.JTS) GO TO 130
   80 WRITE (6,200)
      DO 90 N1=1,NEM
      N=IELEM(M,N1)
      IF (IPLUSO(N)+IPLUS(N).EQ.0.AND.IPLUS(N).NE.0) WRITE (6,270) N
   90 CONTINUE
      WRITE (6,210) J,MCOUNT,NKEEP,ICHCK,TEMP(J),ZFM(J)
      DO 100 N1=1,NEM
      N=IELEM(M,N1)
      KN=KMAX(N)
  100 WRITE (6,220) NAMEL(N),ZF1(N)
      WRITE (6,230) (NI(K,N),K=1,KN)
      WRITE (6,240) (R(K,N),K=1,KN)
      IF (ICHCK.EQ.0) GO TO 110
      GO TO 130
  110 CALL OPAGUE
      GO TO 130
  120 CALL OPAGC
  130 CONTINUE
      IF (ICHCK) 140,150,140
C
C     CALCULATE ENERGY TRANSMISSION ACROSS MESH BOUNDARIES
C
C     IF IFLOW EQUALS 0 NO HEAT TRANSFER RATES ARE BEING CALCULATED
C             THIS CYCLE.
C     IF ITER(J) = 0 NO ENERGY TRANSMISSION AT ZONE BOUNDARY IS
C             CALCULATED THIS CYCLE.
C     IF ITER(J) = 1 TRANSMISSION AT BOUNDARY DUE TO DIFFUSION ONLY
C             IS CALCULATED.
C     IF ITER(J) = 2 TRANSMISSION AT BOUNDARY DUE TO CONDUCTION ONLY
C             IS CALCULATED
C     IF ITER(J) = 3 TRANSMISSION AT BOUNDARY DUE TO DIFFUSION AND
C             CONDUCTION IS CALCULATED.
C     IF ITER(J) EQUALS 4 THEN IFLOW EQUALS ZERO ALSO
C
```

```
  140 IF (IFLOW.EQ.0) GO TO 180
      CALL TRANSP
      IF (J1.EQ.JFIN) GO TO 180
      XLAM1(J1+1)=0.0
      XLAM2(J1+1)=0.0
      GO TO 180
  150 IF (NKEEP/NPRIN*NPRIN.NE.NKEEP) GO TO 180
      NKEAP=NKEEP+1
      DO 170 NS=1,NSPEC
      IF (SS(2,NS)) 170,170,160
  160 WRITE (6,250) NKEAP,NS
      WRITE (6,260) (SS(J,NS),J=1,JFIN)
  170 CONTINUE
      IF (IFLOW.EQ.0) GO TO 180
      WRITE (6,280)
      WRITE (6,260) (F(J),J=1,JFIN)
C     WRITE (6,200)
C     WRITE (6,270) (XFX(J),J=1,JFIN)
  180 N=NKEEP
      RETURN
C
  190 FORMAT (7H0FLOION,2(1X,A10),5H KN =,I3,4H L =,I3,4H J =,I3,8H EI(J
     1) =,1PE10.3,9H R(K,N) =,1PE10.3,10H EN(N,L) =,1PE10.3/10X,10H AF(M
     2,N) =,1PE10.3/)
  200 FORMAT (//)
  210 FORMAT (30H CALCULATED, FLOION SUBROUTINE/10X,4HJ = ,I5,10X,9HMCOU
     1NT = ,I5,10X,8HCYCLE = ,I5,10X,7HICHCK =,I5/10X,13HTEMPERATURE =
     2 ,E10.3,10X,12HIONIZATION =,E10.3/)
  220 FORMAT (15H0IONIZATION IN ,A10,2H= ,E10.3)
  230 FORMAT (23H0IONIZATION LEVELS ARE ,I10,7(1H,,I10))
  240 FORMAT (28H0FRACTIONAL POPULATIONS ARE ,E10.3,7(1H,,E10.3))
  250 FORMAT (//26H DEPOSITION EDIT FOR CYCLE,I10,2X,10HAND SOURCE,I10/2
     1 7H UNITS ARE ERGS/GRAM/SECOND/)
  260 FORMAT (1P10E12.4)
  270 FORMAT (56H THERE IS AN UNCONTROLLED JUMP IN IONIZATION IN ELEMENT
     1 ,I2)
  280 FORMAT (//57H FLUXES INTO THE MESHES IN UNITS OF ERGS/SQUARE CM/SE
     1COND)
      END
```

```
*DECK GENRAT
      SUBROUTINE GENRAT
*IF DEF,B32
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
*ENDIF
C
C                *---*---*---*---*---*---*---*---*---*---*
C
C        IMPORTANT VARIABLES LOCAL TO THIS SUBROUTINE
C
C        CALPA           CALORIES ABSORBED IN ZONE
C        DX              USED AS FIRST ZONE SIZE IN INPUT
C        EITOT           THE TOTAL SOURCE STRENGTH*COSINE OF ANGLE OF
C                        INCIDENCE COMPUTED FOR EACH SPECTRUM.
C        EPG             ERG/GRAM ABSORBED IN ZONE
C        EPREV           USED IN CALCULATIONS OF ENERGY OF X-RAY EDGES
C                        AND IONIZATION POTENTIALS TO STORE THE ENERGY
C                        OF THE PREVIOUS NUCLEUS-ELECTRON CONFIGURATION
C        ERGPA           ERGS ABSORBED IN ZONE
C        EOI             THE ENERGY OF A NUCLEUS-ELECTRON CONFIGURATION
C        NRZC        ZONE CONTROL.  IF POSITIVE, NRZC = THE NUMBER OF
C                    REGIONS FOR WHICH A ZONE SIZE RATIO AND NUMBER OF
C                    ZONES WILL BE SPECIFIED.  IF NEGATIVE, NRZC IS A
C                    FLAG TO INDICATE THAT THE ZONE SIZES ARE TO BE
C                    CALCULATED WITHIN THE PROGRAM.
C        SCREEN          TEMPORARY STORAGE FOR THE SLATER SCREENING
C                        CONSTANTS -- ONE FOR EACH S+P, D, AND F
C                        SUB-SHELL
C        SUMCAL          TOTAL CALORIES/CM**2 ABSORBED UP TO AND
C                        INCLUDING THIS ZONE
C
C                *---*---*---*---*---*---*---*---*---*---*
C
*CALL BLANK
*CALL AA
*CALL AB
*CALL AC
*CALL EQVP
*CALL PLOTCM
*CALL RZCOM
*CALL SPLLC
      DIMENSION ASAVE(5), NSAVE(15), SCREEN(14)
C
      REAL LGDEL
C
C        READ PROBLEM DESCRIPTION, PROBLEM CONTROL VARIABLES, AND
C        OUTPUT CONTROL VARIABLES.
C
      READ (5,1740,END=10) (DISCPT(I),I=1,8)
```

```
   1C READ (5,*,END=20) NSPEC,NTEDT,NJEDIT,LOZHIZ,ILOG,ILIN,ICON,IDIF
      WRITE (6,1920)
      WRITE (6,1740) (DISCPT(K),K=1,8)
      WRITE (7,1740) (DISCPT(K),K=1,8)
      WRITE (9,1740) (DISCPT(K),K=1,8)
      WRITE (9,2170)
      WRITE (6,1760)
      WRITE (6,2040) ICON,IDIF
   20 IF (NTEDT) 40,40,30
C     FORMAT WAS 8E10
   30 READ (5,*,END=40) (TEDIT(I),I=1,NTEDT)
   40 IF (NJEDIT) 60,6C,50
C    FORMAT WAS 8I10
   50 READ (5,*,END=60) (MTLN(I),DSTF(I),I=1,NJEDIT)
C    FORMAT WAS 8I10
   60 READ (5,*,END=70) NRZC,NMTRLS,JRZL,JZPUL,NPRIN,NTAPE,NREZON,JCYCS
C
C     SET DEFAULT VALUES FOR JRZL, JZPUL, NPRIN, NTAPE, NREZON.
C
   70 IF (JRZL.EQ.0) JRZL=25
      IF (JZPUL.EQ.0) JZPUL=50
      IF (NPRIN.EQ.0) NPRIN=20
      IF (NTAPE.EQ.0) NTAPE=100
C
C     DEFAULT FOR NREZON WILL INHIBIT REZONING.
C
      IF (NREZON.EQ.0) NREZON=JCYCS
      NREZON=MAX0(NREZON,25)
      WRITE (6,1780) NRZC,NMTRLS,JRZL,JZPUL,NPRIN,NTAPE,NDEP,LOZHIZ,JFIN
     1 ,JCYCS,NREZON
      READ (5,*,END=80) RSCRIT,RZCO,RZC1
   80 IF (RSCRIT.EQ.0.) RSCRIT=1./FLOAT(JZPUL)
      IF (RZCO.EQ.0.) RZCO=1./FLOAT(JRZL)
      WRITE (6,1790) RSCRIT,RZCO,RZC1
C     FORMAT WAS 8E10
      READ (5,*,END=90) CKS,TS,ANGLE,DTMIN,DIFTST,NDEP,JPRIN,ION
C
C        READ MATERIAL ZONING CONSTANTS AND INITIAL HYDRO TIME STEP.
C
   90 IF (NRZC.LE.0) GO TO 140
      NMT=NMTRLS-1
      IF (NMT) 110,110,100
C    FORMAT WAS 8I10
  100 READ (5,*,END=110) (JBND(M),M=1,NMT)
C    FORMAT WAS 8I10
  110 READ (5,*,END=120) JFIN,(NZ(L),L=1,NRZC)
C     FORMAT WAS 8E10
  120 READ (5,*,END=130) DX,TIME,(RZ(M),M=1,NRZC)
  130 GO TO 160
  140 CONTINUE
```

```
      NRZC=0
      READ (5,*,END=150) TIME,(RZ(M),M=1,NMTRLS)
  150 DX=0
  160 CONTINUE
C
C     THIS CHANGE BY DLJ 2 JUN 88 TO ALLOW FOR DELAYED DEBUG OUTPUT
C     IF JPRIN = 0, NO DEBUG -- SAME AS BEFORE
C     IF DEBUG WANTED, INPUT JPRIN = CYCLE NUMBER AT WHICH THE DEBUG
C     PRINTOUT IS TO START.
C
      NDBG=JPRIN
      IF (NDBG.LE.0) NDBG=JCYCS+1
      IF (JPRIN.GE.1) JPRIN=1
      WRITE (6,1840) ANGLE,TIME,CKS,TS,NDEP
      DTN=TIME
      DTNH=TIME
      DX=DX/RZ(1)
C
C        READ NUMBER OF ELEMENTS IN THIS PROBLEM AND NUMBER OF X-RAY
C        ABSORPTION EDGES TO BE INPUT FOR EACH.
C
C    FORMAT WAS 8I10
      READ (5,*,END=170) NELT
C    FORMAT WAS 8I10
  170 READ (5,*,END=180) (NOE(N),N=1,NELT)
C
C        READ IN DATA FOR EACH ELEMENT AND MAKE NECESSARY INITIAL
C        CALCULATIONS FOR EACH
C
  180 DO 910 N=1,NELT
C     FORMAT WAS 2I10,A10,E10
      READ (5,1710,END=190) NAMEL(N)
  190 READ (5,*,END=200) NTBL(N),NVARE(N),XAW(N)
  200 IT=NOE(N)
C     FORMAT WAS 8E10
      READ (5,*,END=210) (AA(N,I),B(N,I),EDGE(N,I),I=1,IT)
  210 IF (NVARE(N)) 250,250,220
  220 IT=NTBL(N)
C     FORMAT WAS 8E10
      READ (5,*,END=230) (XI(N,J),J=1,IT)
  230 EN(N,1)=XI(N,1)
      DO 240 J=2,IT
  240 EN(N,J)=EN(N,J-1)+XI(N,J)
      NOEC(N)=MIN0(NOE(N),19)
      GO TO 910
C
C     ALL POSSIBLE EDGES OF AN ELEMENT MUST BE INPUT FOR THIS ROUTINE
C        TO OPERATE PROPERLY.
C
C     THIS ROUTINE CALCULATES THE ENERGIES OF THE ABSORPTION EDGES OF
```

```
C        THE ELEMENTS -- EDGEC(ELEMENT,EDGE), THE ENERGY OF ALL POSSIBLE
C        IONIZATION STATES OF THE ELEMENTS -- ENIK(ELEMENT,ION), THE
C        IONIZATION POTENTIAL OF EACH IONIZATION STATE --
C        XI(ELEMENT,ION), AND THE ENERGY USED IN IONIZATION TO THIS
C        STATE.
C
  250 ESUM=0.
      IBB=0
      K1=NTBL(N)
      DO 260 K=1,15
      DO 260 KK=1,14
      EBB(N,K,KK)=0.0
  260 CONTINUE
      DO 890 K2=1,K1
      IF (K2.NE.1.AND.ION.EQ.0) GO TO 550
      DO 270 K=1,14
  270 NSPDF(K)=0
      DO 280 K=1,19
      NSUM(K)=0
  280 NGRUP(K)=0
      NELEC=K1+1-K2
C
C     FIND THE NUMBER OF ELECTRONS IN EACH SUB-SHELL FOR THIS ION STATE
C     WHEN ION EQUALS 1...
      IF (NELEC-78) 310,310,290
  290 DO 300 K=1,6
  300 NGRUP(K+13)=ITBL(NELEC,K)
      GO TO 450
  310 IF (NELEC-69) 340,340,320
  320 DO 330 K=1,6
  330 NGRUP(K+10)=ITBL(NELEC,K)
      GO TO 470
  340 IF (NELEC-46) 370,370,350
  350 DO 360 K=1,6
  360 NGRUP(K+9)=ITBL(NELEC,K)
      GO TO 480
  370 IF (NELEC-28) 400,400,380
  380 DO 390 K=1,6
  390 NGRUP(K+6)=ITBL(NELEC,K)
      GO TO 500
  400 IF (NELEC-10) 430,430,410
  410 DO 420 K=1,6
  420 NGRUP(K+3)=ITBL(NELEC,K)
      GO TO 520
  430 DO 440 K=1,6
  440 NGRUP(K)=ITBL(NELEC,K)
      GO TO 540
  450 DO 460 K=1,3
  460 NGRUP(K+10)=ILTBL5(K)
  470 NGRUP(10)=ILTBL4(1)
  480 DO 490 K=1,3
  490 NGRUP(K+6)=ILTBL3(K)
  500 DO 510 K=1,3
  510 NGRUP(K+3)=ILTBL2(K)
  520 DO 530 K=1,3
  530 NGRUP(K)=ILTBL1(K)
  540 CONTINUE
      GO TO 580
C     CALCULATE THE NUMBER OF ELECTRONS IN EACH SUB-SHELL FOR THIS ION
C     STATE WHEN ION EQUALS 0 ...
  550 NELEC=K1+1-K2
      DO 560 K=1,19
      IF (NGRUP(20-K)) 560,560,570
  560 CONTINUE
  570 NGRUP(20-K)=NGRUP(20-K)-1
C
C     CALCULATE THE NUMBER OF ELECTRONS IN EACH OF THE SUB-SHELL
C        GROUPS NEEDED IN SCREENING CONSTANT CALCULATIONS --
C
  580 NSPDF(1)=NGRUP(1)
      NSPDF(2)=NGRUP(2)+NGRUP(3)
      NSPDF(3)=NGRUP(4)+NGRUP(5)
      NSPDF(4)=NGRUP(6)
      NSPDF(5)=NGRUP(7)+NGRUP(8)
      NSPDF(6)=NGRUP(9)
      NSPDF(7)=NGRUP(10)
      NSPDF(8)=NGRUP(11)+NGRUP(12)
      NSPDF(9)=NGRUP(13)
      NSPDF(10)=NGRUP(14)
      NSPDF(11)=NGRUP(15)+NGRUP(16)
      NSPDF(12)=NGRUP(17)
      NSPDF(13)=NGRUP(18)
      NSPDF(14)=NGRUP(19)
C
C     CALCULATE THE TOTAL NUMBER OF ELECTRONS UP TO AND INCLUDING
C        THOSE IN EACH SUB-SHELL...
C
      NSUM1=0
      DO 590 K=1,19
      NSUM1=NSUM1+NGRUP(K)
  590 NSUM(K)=NSUM1
C
C        FIND THE NUMBER OF SUB-SHELLS ASSOCIATED WITH THIS ATOM (K2=1)
C        OR ION (K2 GREATER THAN 1) -- IN CASE OF THE ATOM STORE THIS
C        NUMBER AS NOEC(N) (NUMBER OF EDGES CALCULATED), ALSO STORE THE
C        NUMBER OF ELECTRONS IN EACH SUB-SHELL AS NION, FOR LATER USE --
C        FOR BOTH ATOM AND ION FIND THE NUMBER OF COMPACTED SUB-SHELLS
C        AND CALL IT K3.
C
      IFLAG=1
      DO 600 K=1,19
      IF (NELEC-NSUM(K)) 610,610,600
  600 CONTINUE
  610 IF (NGRUP(K).NE.1) GO TO 620
      IF (K.NE.2.AND.K.NE.4.AND.K.NE.7.AND.K.NE.11.AND.K.NE.15.AND.K.NE.
     1 19) GO TO 620
      IFLAG=2
      IBB=IBB+1
      NSAVE(IBB)=K2
      NLEC(N,IBB)=K-1
  620 IF (K2-1) 630,630,650
  630 NOEC(N)=K
      KN=NOEC(N)+1
      DO 640 KK=1,K
      IJK=KN-KK
  640 NION(N,IJK)=NGRUP(KK)
      WRITE (6,1680) (NION(N,KK),KK=1,19)
      KN=ITABL(KN-1)
  650 K3=ITABL(K)
      K31=K3+1
      IF (K2.GT.1) K31=1
      IF (IFLAG.EQ.2.AND.K2.NE.1) K31=K3
C
C     FIND THE SCREENING CONSTANTS FOR THE SUB-SHELLS OF THIS ATOM OR
C        ION WITH NO ABSORPTION EDGE ELECTRONS REMOVED -- USE THESE
C        CONSTANTS TO CALCULATE THE ENERGY ASSOCIATED WITH THIS STATE
C        OF THE ELEMENT -- STORE AS EION(ELEMENT,IONIZATION).  ALSO,
C     FIND THE SCREENING CONSTANTS FOR THE ATOM STATE OF THIS ELEMENT
C        WHEN AN ABSORPTION EDGE ELECTRON HAS BEEN REMOVED -- USE
C        THESE TO FIND THE ENERGY OF THIS ATOM AND THUS THE ENERGIES
C        OF THE ABSORPTION EDGES.
C
      DO 850 NG=1,K31
      IF (NG-1) 670,670,660
  660 NSTMP=NSPDF(NG-1)-1
      NSPDF(NG-1)=MAXO(NSTMP,0)
C
C     NG-1 IS THE SUB-SHELL ASSOCIATED WITH THIS ABSORPTION EDGS.
C
C     CALCULATE THE SCREENING CONSTANTS
C
  670 NPREVT=NSPDF(1)+NSPDF(2)
      IF (K3.GE.5) GO TO 680
      GO TO (720,710,700,690), K3
      CALL GOTOER
  680 NPREVS=NSPDF(3)+NSPDF(4)
      SCREEN(5)=FLOAT(K1-NPREVT)-0.85*FLOAT(NPREVS)-0.35*FLOAT(MAXO
     1 (NSPDF(5)-1,0))
  690 SCREEN(4)=FLOAT(K1-NPREVT-NSPDF(3))-0.35*FLOAT(MAXO(NSPDF(4)-1,0))
  700 SCREEN(3)=FLOAT(K1-NSPDF(1))-0.85*FLOAT(NSPDF(2))-0.35*FLOAT(MAXO
     1 (NSPDF(3)-1,0))
  710 SCREEN(2)=FLOAT(K1)-0.85*FLOAT(NSPDF(1))-0.35*FLOAT(MAXO(NSPDF(2)-
     1 1,0))
  720 SCREEN(1)=FLOAT(K1)-0.3*FLOAT(MAXO(NSPDF(1)-1,0))
      IF (K3.LE.5) GO TO 750
      NPREVT=NPREVT+NPREVS
      NPREVS=NSPDF(5)
      KK=5
  730 DO 740 K4=1,2
      K5=KK+K4
      SCREEN(K5)=FLOAT(K1-NPREVT-NPREVS)-0.35*FLOAT(MAXO(NSPDF(K5)-1,0))
      NPREVS=NPREVS+NSPDF(K5)
      IF (K3.LE.K5) GO TO 750
  740 CONTINUE
      KK=KK+3
      SCREEN(KK)=FLOAT(K1-NPREVT)-0.85*FLOAT(NPREVS)-0.35*FLOAT(MAXO
     1 (NSPDF(KK)-1,0))
      NPREVT=NPREVT+NPREVS
      NPREVS=NSPDF(KK)
      IF (KK.LT.K3) GO TO 730
C
C        CALCULATE THE ENERGY OF THIS ATOM OR NUCLEUS-ELECTRON
C        CONFIGURATION.
C
  750 EOI=0.0
      DO 760 K=1,K3
  760 EOI=EOI+SCREEN(K)**2*FLOAT(NSPDF(K))/XNSTAR(K)
      EOI=13.56*EOI
      IF (NG.GT.1) GO TO 790
      IF (K2.EQ.1) GO TO 770
C
C     CALCULATE AND STORE THE VALUES OF THE IONIZATION POTENTIALS --
C        XI(ELEMENT,IONIZATION) -- AND THE ENERGY USED IN IONIZATION --
C        EN(ELEMENT,IONIZATION) -- AT THIS ION STATE.
C
      XI(N,K2-1)=EPREV-EOI
      ESUM=ESUM+XI(N,K2-1)
      EN(N,K2-1)=ESUM
      EION(N,K2-1)=EOI*1.0E-3
  770 EPREV=EOI
      IF (K2.GT.1) GO TO 850
      DO 780 K=1,K3
      K4=K3+1-K
  780 SCRENO(N,K4)=SCREEN(K)
      GO TO 850
C
C     CALCULATE THE ENERGIES OF THE ABSORPTION EDGES AND STORE THEM
C        AS EDGEC(ELEMENT,EDGE) AND STORE THE VALUES OF THE SCREENING
C        CONSTANTS AND ELECTRON NUMBERS OF THESE SUB-SHELLS OF THE
C        ATOM FOR LATER USE...
```

```
C
  790 IF (K2.NE.1) GO TO 800
      EDGEC(N,NG-1)=(EPREV-EOI)*1.E-3
  800 IF (IFLAG.EQ.1.OR.NG.GT.K3) GO TO 840
      KTMP=1
      NTMP=NLEC(N,IBB)+1
      IF (NTMP.EQ.19.OR.NTMP.EQ.15.OR.NTMP.EQ.11) KTMP=4
      IF (NTMP.EQ.7) KTMP=3
      IF (NTMP.EQ.4) KTMP=2
      LTMP=K3
      LLBND=K3+1-KTMP
      LUBND=K3-1
      IF (LLBND.EQ.K3) GO TO 820
      DO 810 K=LLBND,LUBND
  810 SCREEN(K3)=SCREEN(K3)-0.15*NSPDF(K)
  820 EBBTMP=0.0
      DO 830 K=1,K3
  830 EBBTMP=EBBTMP+SCREEN(K)**2*FLOAT(NSPDF(K))/XNSTAR(K)
      K4=KN+2-NG
      EBB(N,IBB,K4)=1.356E-2*EBBTMP
  840 NSPDF(NG-1)=NSTMP+1
  850 CONTINUE
      IF (K2-1) 860,860,890
C
C         REARRANGE THE NUMBERING OF THE EDGES SO THAT THE K EDGE
C         OCCURS AS THE LARGEST NUMBERED OF THE EDGES AND THE S+P
C         SUB-SHELL EDGES ARE ASSIGNED DOUBLY TO BOTH S AND P SUB-SHELL
C         GROUPS.
C
  860 K4=NOEC(N)
      KL=ITABL(K4)
      DO 870 K=1,KL
  870 SCREEN(K)=EDGEC(N,K)
      KL=K4+1
      DO 880 K=1,K4
      KK=ITABL(K)
      IJK=KL-K
  880 EDGEC(N,IJK)=SCREEN(KK)
  890 CONTINUE
C
C     CALCULATE AND STORE THE LAST XI, EN, AND EION...
C
      XI(N,K1)=EPREV
      EN(N,K1)=EPREV
      IF (K1.GE.2) EN(N,K1)=EN(N,K1)+EN(N,K1-1)
      EION(N,K1)=0.
      IF (JPRIN.GT.0) WRITE (6,1690) N,(EION(N,K),K=1,20)
C
C     CALCULATE AND STORE THE VALUES OF BOUND-BOUND TRANSITION ENERGIES
C
```

```
      DO 900 K=1,15
      DO 900 KK=1,14
      IF (EBB(N,K,KK).EQ.0) GO TO 900
      N1=NSAVE(K)
      EBB(N,K,KK)=EION(N,N1)-EBB(N,K,KK)
  900 CONTINUE
C
  910 CONTINUE
C
C        READ IN DATA FOR EACH MATERIAL
C
      DO 1030 M=1,NMTRLS
C     FORMAT WAS A10,2I10
      READ (5,1710.END=920) MATL(M)
  920 READ (5,*,END=930) NVARM(M),NATOM(M)
C     FORMAT WAS 8E10
  930 READ (5,*,END=940) RHO(M),EQSTC(M),EQSTD(M),EQSTE(M),EQSTG(M)
     1 ,EQSTH(M),EQSTS(M),PMIN(M),ISPLLM(M),EM(M)
  940 IF (EM(M).LE.1.E7) EM(M)=4.186E7*EM(M)
C
C     EM IS THE MELT ENERGY OF THE MATERIAL (ERGS/GM)
C
C      ISPLLM IS A FLAG TO INDICATE WHICH MODEL PMIN AND TSPALL ARE
C      BASED ON.
C
C     FORMAT WAS 8E10
      READ (5,*,END=950) CUSP1(M),CUSPA(M),CUSPC(M),CUSPD(M),CUSPG(M)
     1 ,CUSPS(M)
C     FORMAT WAS 8E10
  950 READ (5,*,END=960) YO(M),AMU(M),YADD(M),YMU(M),XMW(M),LGDEL(M)
     1 ,XCON(M)
  960 E(2)=7.5E10*FLOAT(NATOM(M))/XMW(M)
      EM(M)=EM(M)+E(2)
      EQSTE(M)=EQSTE(M)+E(2)
      EQSTN(M)=EQSTC(M)/EQSTG(M)/(EQSTE(M)*RHO(M))
C
C     READ IN PARAMETERS FOR ELASTIC VISCOPLASTIC MODEL AND POSSIBLY
C     FOR MAXWELL GEOMETRIC DISPERSION MODEL.
C
      READ (5,*,END=970) EQSTA(M),OMEGA(M),PRELAX(M),SHEARR(M),TRELAX(M)
     1 ,MFLAG(M)
C
C     IF MFLAG = 1, READ IN PARAMETERS FOR MAXWELL GEOMETRIC
C     DISPERSION MODEL.
C
  970 AMU2(M)=0.
      TRELX2(M)=0.
      IF (MFLAG(M).EQ.1) READ (5,*,END=980) AMU2(M),TRELX2(M)
  980 CH(M)=SQRT(EQSTC(M)/RHO(M))
      GOKE(M)=AMU(M)/EQSTC(M)
      GOKE2(M)=AMU2(M)/EQSTC(M)
C    FORMAT WAS 8I10
      READ (5,*,END=990) NELEM(M)
  990 IF (NELEM(M).LE.5) GO TO 1000
      WRITE (6,1700) NELEM(M),MATL(M)
      STOP
 1000 CONTINUE
      IT=NELEM(M)
C     FORMAT WAS 8I10
      READ (5,*,END=1010) (IELEM(M,N),N=1,IT)
C     FORMAT WAS 8E10
 1010 READ (5,*,END=1020) (ASAVE(N),N=1,IT)
 1020 DO 1030 N=1,IT
      N1=IELEM(M,N)
 1030 AF(M,N1)=ASAVE(N)
C
C         CALCULATE ZONING
C
 1040 CONTINUE
C
C          NRZC IS USED AS A FLAG TO INDICATE WHETHER ZONING IS TO BE
C          CALCULATED BY EXPLICIT INSTRUCTIONS OR AUTOMATICALLY.
C          NRZC .GT. 0 INDICATES ZONING IS DONE PER EXPLICIT INSTRUCTIONS
C          NRZC .EQ. 0 INDICATES THIS IS THE FIRST GUESS AT AUTOMATIC
C          ZONING. .NRZC .LT. 0 INDICATES THIS IS THE REVISED ZONING.
C
      IF (NRZC.LE.0) CALL AZONE (DX,NRZC)
      LZ=1
      DO 1050 J=2,JFIN
      IF (J.GT.NZ(LZ)) LZ=LZ+1
      DX=DX*RZ(LZ)
      X(J)=X(J-1)+DX
      IF (NRZC.EQ.0) GO TO 1050
      PSMAX(J)=0.
      PSMIN(J)=0.
 1050 CONTINUE
      IF (NJEDIT.LE.0) GO TO 1080
      IF (NRZC.LE.0) GO TO 1080
      I=1
      DO 1070 J=2,JFIN
      IF (I.GT.NJEDIT) GO TO 1080
      M=MTLN(I)
      IF (M.GE.2) GO TO 1060
      ASAVE(I)=DSTF(I)*X(JBND(M))
      IF (X(J).LE.ASAVE(I)) JEDIT(I)=J
      IF (X(J).GE.ASAVE(I)) I=I+1
      GO TO 1070
 1060 CONTINUE
      K=M-1
      ASAVE(I)=X(JBND(K))+DSTF(I)*(X(JBND(M))-X(JBND(K)))
      IF (X(J).LE.ASAVE(I)) JEDIT(I)=J
      IF (X(J).GE.ASAVE(I)) I=I+1
 1070 CONTINUE
 1080 CONTINUE
C
C        PRINT ALL NON-ENERGY SOURCE DEPENDENT DATA
C
      WRITE (6,1800)
      K=IABS(NRZC)
      K=MAXO(K,2*NMTRLS-1)
      WRITE (6,1810) (RZ(I),NZ(I),I=1,K)
      IF (NRZC.EQ.0) GO TO 1290
      IF (NJEDIT) 1100,1100,1090
 1090 WRITE (6,1940) (JEDIT(I),I=1,NJEDIT)
 1100 IF (NTEDT) 1110,1120,1110
 1110 WRITE (6,1930) (TEDIT(I),I=1,NTEDT)
 1120 WRITE (6,1970) NELT
      WRITE (6,1960)
C
      DO 1200 N=1,NELT
      WRITE (6,1960)
      WRITE (6,1980) NAMEL(N),N
      IF (NVARE(N)) 1130,1130,1140
 1130 WRITE (6,2090)
      GO TO 1150
 1140 WRITE (6,2100)
 1150 IT=NOE(N)
      WRITE (6,1890) IT,(AA(N,I),B(N,I),EDGE(N,I),I=1,IT)
      IF (NVARE(N)) 1160,1160,1170
 1160 WRITE (6,2050)
      IT=NOEC(N)
      WRITE (6,2060) IT,(EDGEC(N,I),I=1,IT)
      WRITE (6,2070)
      GO TO 1180
 1170 WRITE (6,2080)
 1180 WRITE (6,1990)
      IT=NTBL(N)
      WRITE (6,2130)
      WRITE (6,1750) (XI(N,J),J=1,IT)
      WRITE (6,2130)
      WRITE (6,2000)
      WRITE (6,2130)
      WRITE (6,1750) (EN(N,J),J=1,IT)
      IF (JPRIN.EQ.0) GO TO 1200
      WRITE (6,2130)
      WRITE (6,2140)
      WRITE (6,2160) (NLEC(N,J),J=1,6)
      WRITE (6,2130)
      WRITE (6,2150)
      DO 1190 I=1,6
 1190 WRITE (6,1750) (EBB(N,I,J),J=1,14)
 1200 CONTINUE
      WRITE (6,2130)
      WRITE (6,1960)
      JB1=1
      DO 1280 M=1,NMTRLS
      WRITE (6,1960)
      IF (JBND(M)) 1220,1210,1220
 1210 JB2=JFIN
      GO TO 1230
 1220 JB2=JBND(M)
 1230 THKNS=X(JB2)-X(JB1)
      WRITE (6,1850) MATL(M),RHO(M),JB1,JB2,THKNS
      IF (NVARM(M)) 1240,1240,1250
 1240 WRITE (6,2110)
      GO TO 1260
 1250 WRITE (6,2120)
 1260 JB1=JB2
      WRITE (6,1860) EQSTC(M),EQSTD(M),EQSTE(M),EQSTG(M),EQSTH(M),EQSTS
     1 (M),EQSTN(M),PMIN(M)
      WRITE (6,1870) CUSP1(M),CUSPA(M),CUSPC(M),CUSPD(M),CUSPG(M),CUSPS
     1 (M)
      WRITE (6,1880) YO(M),AMU(M),YADD(M),YMU(M),EQSTA(M),OMEGA(M)
     1 ,PRELAX(M),SHEARR(M),TRELAX(M),MFLAG(M),AMU2(M),TRELX2(M)
      WRITE (6,2010) XCON(M),XMW(M),NATOM(M)
      IT=NELEM(M)
      WRITE (6,2020) (IELEM(M,N),N=1,IT)
      NKEEP=1
      DO 1270 N=1,10
      IF (AF(M,N).EQ.0) GO TO 1270
      ASAVE(NKEEP)=AF(M,N)
      NKEEP=NKEEP+1
 1270 CONTINUE
      WRITE (6,2030) (ASAVE(N),N=1,IT)
 1280 CONTINUE
      WRITE (6,1920)
      IF (NRZC.LT.0) GO TO 1440
 1290 CONTINUE
C
C          READ ENERGY SOURCE DATA
C
      ANGLE=COS(ANGLE/57.2957795)
      DO 1430 NS=1,NSPEC
C     FORMAT WAS 8I10
      READ (5,*,END=1300) NHNU(NS),NBB(NS)
 1300 NNU=NHNU(NS)
      NBBS=NBB(NS)
C     FORMAT WAS 8E10
      READ (5,*,END=1310) START(NS),SSTOP(NS),(T(KK),EE(KK),KK=1,NBBS)
     1 ,XPRIN
 1310 CONTINUE
*IF DEF,B64
      SSTOPM=AMAX1(SSTOPM,SSTOP(NS))
*ENDIF
*IF DEF,B32
      SSTOPM=DMAX1(SSTOPM,SSTOP(NS))
*ENDIF
      IF (NNU) 1320,1350,1320
C
C         FOR ARBITRARY SPECTRUM
C
C     FORMAT WAS 8E10
 1320 CONTINUE
      READ (5,*,END=1330) (TBL(I),ES(NS,I),I=1,NNU)
 1330 EITOT=0.
      DO 1340 II=1,NNU
      ES(NS,II)=ES(NS,II)*ANGLE*EE(1)
      EITOT=EITOT+ES(NS,II)
 1340 CONTINUE
      GO TO 1390
C
C         FOR BLACK BODY SPECTRUM
C
 1350 EITOT=0.
      DO 1380 KK=1,NBBS
      DO 1380 I=1,109
      IF (I-99) 1360,1360,1370
 1360 EITOT=EITOT+EE(KK)*ANGLE*.01
      GO TO 1380
 1370 EITOT=EITOT+EE(KK)*ANGLE*.001
 1380 CONTINUE
C
C         ENERGY INPUT EDIT
C
 1390 IF (NNU) 1410,1400,1410
 1400 WRITE (6,1770) (TBL(I),I=1,109)
      GO TO 1420
 1410 IF (XPRIN.EQ.1.0) GO TO 1430
      WRITE (6,1770) (TBL(I),I=1,NNU)
 1420 WRITE (6,1960)
      WRITE (6,1950) EITOT,NBBS,START(NS),SSTOP(NS)
      WRITE (6,1820)
      WRITE (6,1830) (T(I),EE(I),I=1,NBBS)
      WRITE (6,1960)
 1430 CONTINUE
      XFX(1)=EITOT
      XFL(1)=EITOT
C
C       INITIALIZE SPALL COUNTERS
C
```

```
      IS=0
      ISM=0
      ISPALL=0
 1440 CONTINUE
C
C        INITIALIZE COUNTERS AND CONSTANTS
C
      NRZ=50
      CO=1.8
      C1=.25
      IT=1
      NTEDT=0
      LINE=0
      PDTPOS=0.
      PDTNEG=0.
      IFLOW=0
      IF (ICON.EQ.0.OR.IDIF.EQ.0) IFLOW=1
      IF (NJEDIT) 1470,1470,1450
 1450 DO 1460 I=1,NJEDIT
 1460 JORG(I)=JEDIT(I)
 1470 CONTINUE
C
C     CHECK INPUT DATA VALUES
C
      IF (NPRIN.GT.0) GO TO 1490
      WRITE (6,1720)
 1480 STOP
 1490 IF (JFIN.GT.1.AND.JFIN.LE.201) GO TO 1500
      WRITE (6,1730)
      GO TO 1480
 1500 CONTINUE
C
C        INITIALIZE ZONE VARIABLES
C
      M=1
      DO 1530 J=2,JFIN
      YOZ(J)=YO(M)
      V(J)=1./RHO(M)
      ZM(J)=(X(J)-X(J-1))*RHO(M)
      TEMP(J)=300.
      E(J)=2.5E8/XMW(M)*TEMP(J)*FLOAT(NATOM(M))
      GOVERK(J)=GOKE(M)
      SD2(J)=0.
      TSPALL(J)=ABS(PMIN(M))
      IF (PMIN(M).EQ.0.) TSPALL(J)=1.E15
C
C      TSPALL IS THE NUMERICAL VALUE OF THE SPALL CRITERIA.
C
      IF (J-JBND(M)) 1530,1510,1530
 1510 CONTINUE
```

```
      IF (NRZC.EQ.0) GO TO 1520
      EM(M)=EM(M)+E(J)
      EQSTE(M)=EQSTE(M)+E(J)
 1520 CONTINUE
      M=M+1
 1530 CONTINUE
      TSPALL(1)=0.
C
C     INITIALIZE SPALL VARIABLES.
C
      DO 1540 I=1,50
      SM(I)=0.
      US(I)=0.
      XS(I)=0.
 1540 CONTINUE
C
C        CALCULATE ENERGY DEPOSITION
C
      N=0
      JHAT=0
      ICHCK=0
      CALL FLOION
      ICHCK=1
      IF (NRZC.NE.0) GO TO 1550
      DX=X(2)+SQRT(EQSTC(1)/RHO(1))*SSTOPM
      NRZC=-1
      GO TO 1040
 1550 CONTINUE
C
C        DEPOSITION EDIT
C
      WRITE (6,1900) (DISCPT(I),I=1,8)
      M=1
      SUMCAL=0.
      DO 1640 J=2,JFIN
      EPG=0.
      DO 1560 I=1,NSPEC
 1560 EPG=SS(J,I)*(SSTOP(I)-START(I))+EPG
      IF (J-(JBND(M)+1)) 1580,1570,1580
 1570 M=M+1
 1580 IF (EPG*EQSTG(M)-1.E7) 1590,1590,1610
 1590 IF (JSTAR) 1600,1600,1610
 1600 JSTAR=J
 1610 IF (4.E-11*EPG*XMW(M).GT.1.2E4) JHAT=J+1
      DX=X(J)-X(J-1)
      ERGPA=EPG*RHO(M)*DX
      CALPA=ERGPA*1.E-7/4.186
      SUMCAL=SUMCAL+CALPA
      WRITE (6,1910) J,DX,X(J),ERGPA,CALPA,SUMCAL,EPG,YOZ(J),ZM(J)
     1 ,TSPALL(J)
```

```
      IF (MOD(J,50)) 1640,1620,1640
 1620 IF (J-JFIN) 1630,1640,1640
 1630 WRITE (6,1900) (DISCPT(I),I=1,8)
 1640 CONTINUE
      JHAT=MINO(JFIN,JHAT)
      IF (JSTAR) 1650,1650,1660
 1650 JSTAR=JFIN
 1660 WRITE (6,1920)
C
C     INITIALIZE PLOT VARIABLES.
C
      DO 1670 I=1,201
      PSMAX(I)=0.0
      PSMIN(I)=0.0
      PX(I)=0.0
      IF (I.GT.JFIN) GO TO 1670
      PX(I)=X(I)
 1670 CONTINUE
      JPRIN=0
      RETURN
C
 1680 FORMAT ('ONION=',19I6)
 1690 FORMAT ('OEION',I5/(1X,1P10E12.4))
 1700 FORMAT (15HOFATAL ERROR***,I5,28H ELEMENTS INPUT FOR MATERIAL,A10,
     1 16H, LIMIT IS 5 ***)
 1710 FORMAT (BZ,A10)
 1720 FORMAT (29HOERROR - NPRIN MUST BE .GT. 0)
 1730 FORMAT (41HOERROR - JFIN MUST BE .GT. 1 AND .LE. 201)
 1740 FORMAT (BZ,8A10)
 1750 FORMAT (8E10.3)
 1760 FORMAT (46H ***** THIS PROBLEM WAS RUN WITH HIRAD *****  /)
 1770 FORMAT (13H TABLE VALUES/(10E10.3,/))
 1780 FORMAT (/,6X,4HNRZC,4X,6HNMTRLS,6X,4HJRZL,5X,5HJZPUL,5X,5HNPRIN,5X
     1 ,5HNTAPE,6X,4HNDEP,4X,6HLOZHIZ,6X,4HJFIN,5X,5HJCYCS,4X,6HNREZON/1
     2 1I10)
 1790 FORMAT (/,14X,'RSCRIT',6X,'RZC0',6X,'RZC1'/15X,1P3E10.3/)
 1800 FORMAT (/12H ZONING USED/)
 1810 FORMAT (3X,6H RATIO,1PE22.13,8H TO ZONE,I4)
 1820 FORMAT (/17H BLACK BODY INPUT/)
 1830 FORMAT (12H TEMPERATURE,5X,7H ENERGY/,10(2E12.3,/))
 1840 FORMAT (/5X,5HANGLE,6X,4HTIME,7X,3HCKS,8X,2HTS,6X,4HNDEP/4E10.3
     1 ,I10)
 1850 FORMAT (/25HCMATERIAL PROPERTIES FOR ,A10/5X,4HRHO=,1PE12.4,5X,7HF
     1ROM J=,I4,1X,5HTO J=,I4,5X,11HTHICKNESS =,1PE12.4)
 1860 FORMAT (/10X,5HEQSTC,10X,5HEQSTD,10X,5HEQSTE,10X,5HEQSTG,10X,5HEQS
     1TH,10X,5HEQSTS,10X,5HEQSTN,11X,4HPMIN/1P8E15.5)
 1870 FORMAT (/10X,5HCUSP1,10X,5HCUSPA,10X,5HCUSPC,10X,5HCUSPD,10X,5HCUS
     1PG,10X,5HCUSPS/1P8E15.5)
 1880 FORMAT (/13X,2HY0,12X,3HAMU,11X,4HYADD,12X,3HYMU,10X,5HEQSTA,10X,5
     1 HOMEGA,9X,6HPRELAX,9X,6HSHEARR/1P8E15.5//9X,6HTRELAX,10X,5HMFLAG,
     2 11X,4HAMU2,9X,6HTRELX2/1PE15.5,I15,1P2E15.5)
1890 FORMAT (/,6H NOE =,I3,19X,2HAA,14X,1HB,11X,4HEDGE/,20(15X,3E15.5/)
     1 )
1900 FORMAT (1H1,8A10/3HO J,5X,3H DX,12X,2H X,10X,5H ERGS,10X,4H CAL,8X
     1 ,8H SUM CAL,6X,8H ERGS/GM,8X,4H YOZ,7X,10H ZONE MASS,6X,7H TSPALL
     2 //)
1910 FORMAT (1H ,I3,1P9E14.4)
1920 FORMAT (1H1)
1930 FORMAT (/15H THE TEDITS ARE/(10E10.3/))
1940 FORMAT (/15H THE JEDITS ARE/10I10/)
1950 FORMAT (5X,5HEITOT,7X,3HNBB,5X,5HSTART,5X,5HSSTOP,/,E10.3,I10,2E10
     1 .3)
1960 FORMAT (//)
1970 FORMAT (//39H NUMBER OF ELEMENTS IN THIS PROBLEM IS ,I10/)
1980 FORMAT (/12H ELEMENT IS ,A10,18HELEMENT NUMBER IS ,I10)
1990 FORMAT (' IONIZATION LEVELS FOR ZERO TO COMPLETE IONIZATION (IN','
     1 EV) ARE ')
2000 FORMAT (' IONIZATION ENERGIES PER ATOM (IN EV) FOR ZERO TO COMPL''
     1ETE IONIZATION ARE ')
2010 FORMAT (/39H ATOM CONDUCTIVITY OF THIS MATERIAL IS ,E10.3/38H MOLE
     1CULAR WEIGHT OF THIS MATERIAL IS ,E10.3/37H NUMBER OF ATOMS IN THI
     2S MATERIAL IS ,I10)
2020 FORMAT (/50H ELEMENT NUMBERS WHICH OCCUR IN THIS MATERIAL ARE /10X
     1 ,7I10)
2030 FORMAT (/38H ATOM FRACTIONS OF THESE ELEMENTS ARE /4X,7E10.3)
2040 FORMAT (/,' IF ICON EQUALS ZERO CONDUCTIVITY IN THE VAPOR STATE
     1 IS ASSUMED , ICON FOR THIS PROBLEM EQUALS ',I10/' IF IDIF EQUALS
     2 ZERO DIFFUSION OF ENERGY IN THE MATERIALS IS ASSUMED , IDIF FOR T
     3HIS PROBLEM EQUALS ',I10/)
2050 FORMAT (//' ENERGY OF THE EDGES WERE CALCULATED BY THIS CODE AS')
2060 FORMAT (/7H NOEC =,I3,37X,5HEDGEC/20(40X,E15.5/))
2070 FORMAT (//' IONIZATION POTENTIALS AND ENERGIES WERE CALCULATED BY
     1 THE CODE FOR THIS ELEMENT')
2080 FORMAT (//' IONIZATION POTENTIALS AND ENERGIES WERE INPUT FOR THIS
     1 ELEMENT')
2090 FORMAT (/' THE ABSORPTION CROSS SECTION OF THIS ELEMENT IS VARIABL
     1E UNLESS THE MATERIAL IN WHICH IT OCCURS DOES NOT ACCEPT VARIABLE'
     2 /X,14HCROSS SECTIONS)
2100 FORMAT (/' THE ABSORPTION CROSS SECTION OF THIS ELEMENT IS FIXED
     1 AT ITS COLD VALUE')
2110 FORMAT (/' THE ABSORPTION CROSS SECTIONS OF THE ELEMENTS IN THIS
     1 MATERIAL ARE VARIABLE UNLESS NVARE(ELEMENT) NOT EQUAL TO ZERO')
2120 FORMAT (/' THE ABSORPTION CROSS SECTIONS OF THE ELEMENTS IN THIS
     1 MATERIAL ARE FIXED AT THEIR COLD VALUES')
2130 FORMAT (/)
2140 FORMAT (37H SUB-SHELLS AT NTMP EQUALS ONE TO SIX)
2150 FORMAT (' BOUND-BOUND TRANSITION ENERGIES FOR NTMP EQUALS ONE TO
     1 SIX')
2160 FORMAT (8I10)
2170 FORMAT (1HO,4X,4HTIME,8X,5HEMVBM,7X,6HEMVBM1,6X,6HEMVBM2,6X,6HEMVN
1EG,6X,6HEMVPOS,7X,5HRERAD//)
 END
```

```
*DECK HYDRO
      SUBROUTINE HYDRO
*IF DEF,B32
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
*ENDIF
C
C         IMPORTANT VARIABLES LOCAL TO THIS SUBROUTINE
C
C         DAVG           THE AVERAGE DENSITY FOR CURRENT AND PREVIOUS
C                        CYCLE
C         DISTE          THE SPECIFIC ENERGY USED THIS CYCLE IN CAUSING
C                        PLASTIC DISTORTION
C         DOLD           THE DENSITY DURING THE PREVIOUS CYCLE
C         DU             THE PARTICLE VELOCITY AT THIS ZONE BOUNDARY
C                        MINUS THAT AT THE PREVIOUS ZONE BOUNDARY
C         EMU            THE QUANTITY ((RHO/RHOO)-1) -- IF LESS THAN ZERO
C                        THE MATERIAL IS CONSIDERED TO BE A VAPOR
C         VELS           THE VELOCITY STRAIN USED IN CALCULATING THE
C                        CHANGE IN DEVIATORIC STRESS IN THE DIRECTION
C                        OF PARTICLE MOTION DURING A HYDRO CYCLE --
C                        EQUAL TO DV*DAVG
C         VMC            THE VON MISES YIELD CRITERION REQUIRES VMC EQUAL
C                        1.5*SD**2 BE LESS THAN 0.667*YIELDSTRENGTH**2
C
C                  *---*---*---*---*---*---*---*---*---*---*
C
*CALL BLANK
*CALL EQVP
*CALL PLOTCM
*CALL RZCOM
*CALL SPLLC
C     DATA STATEMENT
C
      DATA ONE, PI, ZERO /1.,3.1415926536,0./
C
C     ENERGY ADDITION FOR ZONES EACH HYDRO-STEP
C
      IF (TIME.GT.SSTOPM) GO TO 90
      DO 80 K=2,JFIN
      EADD(K)=0.0
      DO 70 I=1,NSPEC
      IF (TIME-START(I)) 70,70,10
   10 IF (TIME-DTNH-SSTOP(I)) 20,70,70
   20 IF (TIME-DTNH-START(I)) 30,30,40
   30 EADD(K)=EADD(K)+SS(K,I)*(TIME-START(I))
      GO TO 70
   40 IF (TIME-SSTOP(I)) 50,50,60
   50 EADD(K)=EADD(K)+SS(K,I)*DTNH
      GO TO 70
   60 EADD(K)=EADD(K)+SS(K,I)*(SSTOP(I)-TIME+DTNH)
   70 CONTINUE
   80 CONTINUE
   90 M=1
      LL=1
      MS=1
      JB(1)=1
C
C     JB(M) = INDEX OF LEFT-HAND BOUNDARY OF THE FIRST ZONE TREATED AS
C             DISPERSIVE, IF THERE IS DISPERSIVE MATERIAL.
C
C        LEFT BOUNDARY CONDITIONS
C
      DUDT1=-2.*(S(2)+Q(2)-S(1))/ZM(2)
      U(1)=U(1)+.5*DTN*DUDT1
      X(1)=X(1)+DTNH*U(1)
C
C        HYDRO ZONE LOOP
C
      DO 430 J=2,JFIN
      DOLD=1./V(J)
      QO(J)=Q(J)
      SDOLD=SD(J)
      YOZOLD=YOZ(J)
C
C        CHANGE MATERIAL INDEX AND ADD NEW ACTIVE ZONE
C
      IF (J-JBND(M)) 110,100,110
  100 LL=LL+1
      JB(LL)=JBND(M)
  110 IF (J.LT.JFIN) GO TO 120
      U(JFIN)=U(JFIN)+DTN*(S(JFIN)+Q(JFIN))/ZM(JFIN)
      GO TO 130
C
C        VELOCITY CALCULATION
C
  120 CONTINUE
C
C     CHECK FOR SPALLED ZONE
C
      IF (TSPALL(J).EQ.1.234) GO TO 150
      DUDTS=-2.0*(S(J+1)+Q(J+1)-S(J)-Q(J))/(ZM(J)+ZM(J+1))
      DUDT2=DUDTS
      U(J)=U(J)+.5*DUDT2*DTN
  130 IF (ABS(U(J)).LT.1.E-3) U(J)=0.0
      DU=U(J)-U(J-1)
      IF (U(J).EQ.ZERO) GO TC 140
      QQQ=ABS(DU/U(J))
      IF (QQQ.LE.1.0E-10) DU=0.0
  140 CONTINUE
```

```
C
C          COORDINATE CALCULATION
C
      X(J)=X(J)+DTNH*U(J)
C
C          DENSITY CALCULATION ACTUALLY SPECIFIC VOLUME
C
      V(J)=(X(J)-X(J-1))/ZM(J)
      GO TO 160
  150 CONTINUE
C
C     VELOCITY, COORDINATE, AND DENSITY FOR SPALLED ZONE
C
      ISPALL=3
      DUDT2=-2.*(S(J+1)+Q(J+1))/ZM(J+1)
      DUDTS=2.*(S(J)+Q(J))/ZM(J)
      U(J)=U(J)+.5*DUDT2*DTN
      US(MS)=US(MS)+.5*DUDTS*DTN
      IF (ABS(U(J)).LT.1.E-3) U(J)=0.
      IF (ABS(US(MS)).LT.1.E-3) US(MS)=0.
      DU=US(MS)-U(J-1)
C
C      COORDINATES OF SPALLED ZONE
C
      X(J)=X(J)+DTNH*U(J)
      XS(MS)=XS(MS)+DTNH*US(MS)
C
C     DENSITY - ACTUALLY, SPECIFIC VOLUME
C
      V(J)=(XS(MS)-X(J-1))/ZM(J)
  160 CONTINUE
      DAVG=(1./V(J)+DOLD)/2.
      DV(J)=DTNH*DU/ZM(J)
C
C     VISCOUS STRESS CALCULATIONS
C
      IF (MFLAG(M).GT.0) GO TO 200
  170 CONTINUE
      IF (DU+1.) 180,190,190
C
C     VISCOUS STRESS FOR NON DISPERSIVE MATERIAL
C
  180 Q(J)=(DU*CO*CO-C1*CS(J))*DU*DAVG
      IF (Q(J)-1.E5) 190,220,220
  190 Q(J)=0.
      DU=0.
      GO TO 220
  200 IF (TSPALL(J).NE.8.) GO TO 210
C
C      IF TSPALL = 8., TREAT MATERIAL AS NON DISPERSIVE IN CALCULATING
```

```
C       VISCOUS STRESS FOR DISPERSIVE MATERIAL
C
      JB(M)=J
      GO TO 170
  210 XX=2.*(RHO(M)*CH(M)/OMEGA(M))**2/ZM(J)
*IF DEF,B64
      QU(J)=-XX*EQSTA(M)*OMEGA(M)*DU+DAVG*CO**2*(AMIN1(DU,ZERO))**2
*ENDIF
*IF DEF,B32
      QU(J)=-XX*EQSTA(M)*OMEGA(M)*DU+DAVG*CO**2*(DMIN1(DU,ZERO))**2
*ENDIF
      Q(J)=-XX*(DUDTS-DUDT1)+QU(J)
C
C         STRESS - STRAIN CALCULATION
C
  220 VELS=DV(J)*DAVG
*IF DEF,B64
      THETA=-SD(J)/AMAX1(ONE,SD(J))
*ENDIF
*IF DEF,B32
      THETA=-SD(J)/DMAX1(ONE,SD(J))
*ENDIF
      BULK=DOLD*CS(J)**2
      STRAIN=1.-RHO(M)*V(J)
      IF (SHEARR(M).EQ.ZERO) GO TO 230
      IF (1./V(J)/RHO(M)-1..LE.YMU(M)) GO TO 240
      GOK=GOVERK(J)*EXP(SHEARR(M)*(1./V(J)-DOLD)/RHO(M))
      GOKOLD=GOVERK(J)
*IF DEF,B64
      GOVERK(J)=AMIN1(GOK,GOKE(M))
*ENDIF
*IF DEF,B32
      GOVERK(J)=DMIN1(GOK,GOKE(M))
*ENDIF
      VAMU(J)=BULK*(GOKOLD+GOVERK(J))/1.5
      GO TO 260
  23C IF (STRAIN.LE.O.) GO TO 250
      VAMU(J)=BULK*GOKE(M)/.75
      IF (VAMU(J)-AMU(M)/.75) 250,260,260
  240 VAMU(J)=BULK*GOVERK(J)/.75
      GO TO 260
  25C VAMU(J)=AMU(M)/.75
  260 CONTINUE
      CKSHR=1.
      VAMU(J)=CKSHR*VAMU(J)
      SD(J)=SD(J)+VAMU(J)*VELS
C
C     MAXWELL GEOMETRIC DISPERSION MODEL
C
      IF (AMU2(M).EQ.O..AND.TRELX2(M).EQ.O.) GO TO 290
      IF (STRAIN.LE.0.) GO TO 270
      VAMU2=BULK*GOKE2(M)/.75
      GO TO 280
  270 VAMU2=AMU2(M)/.75
  280 SD2OLD=SD2(J)
      SD2(J)=SD2(J)+VAMU2*VELS
      IF (TRELX2(M).LE.0.) GO TO 290
      TRAT=0.5*DTNH/TRELX2(M)
      SD2(J)=(SD2(J)-TRAT*SD2OLD)/(1.+TRAT)
  290 CONTINUE
      VMC=1.5*SD(J)*SD(J)
      YOZDEG=CKSHR*YOZ(J)
      IF (VMC-YOZDEG*YOZDEG/1.5) 350,350,300
  300 YOZ(J)=YOZ(J)+YADD(M)*ABS(DV(J))*DOLD/RHO(M)/(.2-YMU(M))/V(J)
      YOZDEG=CKSHR*YOZ(J)
      IF (VMC-YOZDEG*YOZDEG/1.5) 350,350,310
  310 IF (TRELAX(M).LE.0.) GO TO 340
      YZDEGO=YOZOLD*CKSHR
      IF (PRELAX(M).EQ.0.) GO TO 320
      TRELAV=TRELAX(M)*EXP(-(ABS(SD(J))-YOZDEG/1.5)/PRELAX(M))
      GO TO 330
  320 TRELAV=TRELAX(M)
  330 SD(J)=(SD(J)-.5*DTNH*(SDOLD+THETA*(YZDEGO+YOZDEG)/1.5)/TRELAV)/(1.
     1 +.5*DTNH/TRELAV)
      GO TO 350
  340 SD(J)=SD(J)*YOZDEG*SQRT(1./VMC/1.5)
  350 IF (STRAIN) 360,380,380
  360 CONTINUE
      IF ((E(J)+EADD(J)).LT.EQSTE(M)) GO TO 370
      SD(J)=0.
      SD2(J)=0.
  370 CONTINUE
  380 DISTE=(SD(J)+SD2(J))*DV(J)
      EADD(J)=EADD(J)+DISTE
      DUDT1=DUDT2
      IF (ISPALL.GE.2) MS=MS+1
      ISPALL=0
      M=LL
C
C        END OF CYCLE CHECKS
C
      IF (U(J)) 430,390,430
  390 IF (N-1) 400,400,410
  400 JSMAXI=JSMAX+10
      IF (LOZHIZ) 410,420,410
C
C        JSTAR CALCULATION
C
  410 IF (J-JSTAR) 430,430,420
  420 JSTAR=J-1
```

```
      GO TO 440
  430 CONTINUE
      JSTAR=JFIN
  440 CONTINUE
C
C     DISPERSIVE MATERIAL CALULATIONS
C
      Q(JSTAR+2)=0.
      DO 550 M=1.LL
      IF (MFLAG(M).EQ.0) GO TO 550
      JL=JB(M)
      JJL=JL+1
      IF (M.EQ.NMTRLS) GO TO 450
      JJU=MINO(JBND(M),JSTAR+1)
      GO TO 460
  450 JJU=JSTAR+1
  460 CSMAX=0.
      SPQMAX=0.
      JSPQM=0
      TPQMAX=0.
      JTPQM=0
      DO 520 J=JJL,JJU
      IF (CS(J).LE.CSMAX) GO TO 470
      CSMAX=CS(J)
  470 IF (TSPALL(J).NE.1.234.AND.J.NE.JJU) GO TO 520
      JU=J
C
C     IMPLICIT SOLUTION FOR Q IN DISPERSIVE MATERIAL
C
      IF (JL.GE.JU) GO TO 510
      YY(JL)=0.
      ZZ(JL)=Q(JL)
      IF (TSPALL(JL).EQ.1.234) ZZ(JL)=0.
      JJL=JL+1
      DO 480 K=JJL,JU
      XX=2.*(RHO(M)*CH(M)/OMEGA(M))**2/ZM(K)
      X1=2./(ZM(K)+ZM(K+1))
      X2=2./(ZM(K-1)+ZM(K))
      IF (K.EQ.JJL.AND.TSPALL(JL).EQ.1.234) X2=2./ZM(K)
      IF (K.EQ.JU.AND.TSPALL(JU).EQ.1.234) X1=2./ZM(K)
      A1=XX*X1
      A2=1.+XX*(X1+X2)
      A3=XX*X2
      A4=QU(K)+XX*(S(K+1)*X1-S(K)*(X1+X2)+S(K-1)*X2)
      IF (K.EQ.JJL.AND.TSPALL(JL).EQ.1.234) A4=A4-XX*S(K-1)*X2
      IF (K.EQ.JU.AND.TSPALL(JU).EQ.1.234) A4=A4-XX*S(K+1)*X1
      A5=A2-A3*YY(K-1)
      YY(K)=A1/A5
      ZZ(K)=(A4+A3*ZZ(K-1))/A5
  480 CONTINUE
```

```
      DO 500 I=JJL,JU
      JJ=JU-I+JJL
      Q(JJ)=YY(JJ)*Q(JJ+1)+ZZ(JJ)
      IF (JJ.EQ.JU.AND.TSPALL(JU).EQ.1.234) Q(JJ)=ZZ(JJ)
C
C     SMAX CALCULATION FOR DISPERSIVE ZONES
C
      IF (S(JJ)+Q(JJ).LE.SPQMAX) GO TO 490
      SPQMAX=S(JJ)+Q(JJ)
      JSPQM=JJ
C
C     TMAX CALCULATION FOR DISPERSIVE ZONES
C
  490 IF (S(JJ)+Q(JJ).GE.TPQMAX) GO TO 500
      TPQMAX=S(JJ)+Q(JJ)
      JTPQM=JJ
  500 CONTINUE
  510 JL=JU
  520 CONTINUE
      IF (SPQMAX.LE.SMAX) GO TO 530
      SMAX=SPQMAX
      JSMAX=JSPQM
  530 IF (TPQMAX.GE.TMAX) GO TO 540
      TMAX=TPQMAX
      JTMAX=JTPQM
  540 IF (JSTAR.GT.JU-1) GO TO 550
C
C     RESET JSTAR IN DISPERSIVE MATERIALS
C
*IF DEF,B64
      JAD=IFIX(CSMAX*DTNH/(X(JSTAR+1)-X(JSTAR)))
*ENDIF
*IF DEF,B32
      JAD=IDINT(CSMAX*DTNH/(X(JSTAR+1)-X(JSTAR)))
*ENDIF
      JSTAR=MIN0(JSTAR+1+JAD,JFIN-1)
  550 CONTINUE
C
      END
```

```
*DECK OPAGUE
      SUBROUTINE OPAGUE
*IF DEF,B32
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
*ENDIF
*CALL BLANK
*CALL AA
*CALL AC
*CALL EQFL
*CALL INDX
      DIMENSION ENIK(10,8,14), ESJ(3,109), ISTART(10,8), NEDGLN(10,8,3),
     1 NNIK(10,8,19), NOEI(10,8), SCREEN(14), SHIFT(10,8,14), SNIK(10,8,
     2 14)
      DATA ISTART, NNIK, ENIK, SHIFT, SNIK, NEDGLN /80*0,1520*0,1120*0.,
     1 1120*0.,1120*1.,240*0/
      DATA SCREEN /14*0./
C
C     THIS PORTION OF THIS ROUTINE CALCULATES THE DEPOSITION OF ENERGY
C        FOR EITHER CONSTANT OR VARIABLE CROSS SECTIONS.
C
      IF (NVARM(M)) 10,10,620
   10 DO 610 N1=1,NEM
      N=IELEM(M,N1)
      KK=NOEC(N)
      KC=ITABL(KK)
      IF (NVARE(N)) 20,20,30
   20 IF (ZF(N,J)-0.5) 30,30,60
C
C     WHEN NVARE(ELEMENT) IS NOT EQUAL TO ZERO, CROSS/SECTIONS ARE
C        FIXED AT COLD VAULE -- THUS SOME CONSTANTS MUST BE SET.
C
   30 DO 40 K=1,14
   40 SNIK(N,1,K)=1.
      DO 50 K=1,19
   50 NNIK(N,1,K)=NION(N,K)
      GO TO 610
C
C     CALCULATE ALL SCREENING CONSTANTS FOR THIS ELEMENT IN EACH OF SIX
C        CALCULATED ION STATES.
C
   60 KN=KMAX(N)
      DO 600 K1=1,KN
      NELEC=NTBL(N)-NI(K1,N)
      JI=NI(K1,N)
      IF (JI.EQ.0.AND.ION.EQ.1) GO TO 570
      IF (JI.GE.NTBL(N)) GO TO 600
      IF (K1.NE.1.AND.ION.EQ.0) GO TO 340
      IF (ION.EQ.0) NELEC=NTBL(N)
C
```

```
C     FIND THE NUMBER OF ELECTRONS IN EACH SUB-SHELL FOR THIS
C     ION STATE IF ION EQUALS 1 OR FOR THE NEUTRAL ATOM IF ION EQUALS 0.
C
      DO 70 K=1,14
   70 NSPDF(K)=0
      DO 80 K=1,19
      NSUM(K)=0
   80 NGRUP(K)=0
      IF (NELEC-78) 110,110,90
   90 DO 100 K=1,6
  100 NGRUP(K+13)=ITBL(NELEC,K)
      GO TO 250
  110 IF (NELEC-69) 140,140,120
  120 DO 130 K=1,6
  130 NGRUP(K+10)=ITBL(NELEC,K)
      GO TO 270
  140 IF (NELEC-46) 170,170,150
  150 DO 160 K=1,6
  160 NGRUP(K+9)=ITBL(NELEC,K)
      GO TO 280
  170 IF (NELEC-28) 200,200,180
  180 DO 190 K=1,6
  190 NGRUP(K+6)=ITBL(NELEC,K)
      GO TO 300
  200 IF (NELEC-10) 230,230,210
  210 DO 220 K=1,4
  220 NGRUP(K+3)=ITBL(NELEC,K)
      GO TO 320
  230 DO 240 K=1,3
  240 NGRUP(K)=ITBL(NELEC,K)
      GO TO 380
  250 DO 260 K=1,3
  260 NGRUP(K+10)=ILTBL5(K)
  270 NGRUP(10)=ILTBL4(1)
  280 DO 290 K=1,3
  290 NGRUP(K+6)=ILTBL3(K)
  300 DO 310 K=1,3
  310 NGRUP(K+3)=ILTBL2(K)
  320 DO 330 K=1,3
  330 NGRUP(K)=ILTBL1(K)
      IF (ION.EQ.1) GO TO 380
C     FIND THE NUMBER OF ELECTRONS IN EACH SUB-SHELL FOR THIS ION STATE
C     ION EQUALS 0
      NELEC=NTBL(N)-NI(K1,N)
  340 NSUM1=0
      DO 350 K=1,19
      NSUM1=NSUM1+NGRUP(K)
      NSUM(K)=NSUM1
      IF (NSUM1.GE.NELEC) GO TO 360
  350 CONTINUE
  360 NGRUP(K)=NELEC-NSUM(K-1)
      KKEEP=K+1
      DO 370 K=KKEEP,19
  370 NGRUP(K)=0
      IF (JI.EQ.0) GO TO 570
C
C     CALCULATE THE NUMBER OF ELECTRONS IN EACH OF THE SUB-SHELL GROUPS
C        NEEDED IN SCREENING CONSTANT CALCULATEIONS --S+P, D, OR F --
C
  380 NSPDF(1)=NGRUP(1)
      NSPDF(2)=NGRUP(2)+NGRUP(3)
      NSPDF(3)=NGRUP(4)+NGRUP(5)
      NSPDF(4)=NGRUP(6)
      NSPDF(5)=NGRUP(7)+NGRUP(8)
      NSPDF(6)=NGRUP(9)
      NSPDF(7)=NGRUP(10)
      NSPDF(8)=NGRUP(11)+NGRUP(12)
      NSPDF(9)=NGRUP(13)
      NSPDF(10)=NGRUP(14)
      NSPDF(11)=NGRUP(15)+NGRUP(16)
      NSPDF(12)=NGRUP(17)
      NSPDF(13)=NGRUP(18)
      NSPDF(14)=NGRUP(19)
C
C     CALCULATE THE TOTAL NUMBER OF ELECTRONS UP TO AND INCLUDING
C        THOSE IN EACH SUB-SHELL
C
      NSUM1=0
      DO 390 K=1,19
      NSUM1=NSUM1+NGRUP(K)
  390 NSUM(K)=NSUM1
C
C     FIND THE NUMBER OF SCREENING CONSTANTS -- OR SUB-SHELLS --
C        ASSOCIATED WITH THIS ATOM OR ION STATE --- CALL IT K3
C
      DO 400 K=1,19
      IF (NELEC-NSUM(K)) 410,410,400
  400 CONTINUE
  410 NOEI(N,K1)=K
      KI=K
      K3=ITABL(K)
      DO 540 K2=1,K3
      NSTMP=NSPDF(K2)-1
      NSPDF(K2)=MAX0(NSTMP,0)
C
C     CALCULATE THE SCREENING CONSTANTS
C
      NPREVT=NSPDF(1)+NSPDF(2)
      IF (K3.GE.5) GO TO 420
      GO TO (460,450,440,430), K3
      CALL GOTOER
  420 NPREVS=NSPDF(3)+NSPDF(4)
      SCREEN(5)=FLOAT(NTBL(N)-NPREVT)-0.85*FLOAT(NPREVS)-0.35*FLOAT(MAXO
     1 (NSPDF(5)-1,0))
  430 SCREEN(4)=FLOAT(NTBL(N)-NPREVT-NSPDF(3))-0.35*FLOAT(MAXO(NSPDF(4)-
     1 1,0))
  440 SCREEN(3)=FLOAT(NTBL(N)-NSPDF(1))-0.85*FLOAT(NSPDF(2))-0.35*FLOAT
     1 (MAXO(NSPDF(3)-1,0))
  450 SCREEN(2)=FLOAT(NTBL(N))-0.85*FLOAT(NSPDF(1))-0.35*FLOAT(MAXO
     1 (NSPDF(2)-1,0))
  460 SCREEN(1)=FLOAT(NTBL(N))-0.3*FLOAT(MAXO(NSPDF(1)-1,0))
      IF (K3.LE.5) GO TO 490
      NPREVT=NPREVT+NPREVS
      NPREVS=NSPDF(5)
      KK=5
  470 DO 480 K4=1,2
      K5=KK+K4
      SCREEN(K5)=FLOAT(NTBL(N)-NPREVT-NPREVS)-0.35*FLOAT(MAXO(NSPDF(K5)-
     1 1,0))
      NPREVS=NPREVS+NSPDF(K5)
      IF (K3.LE.5) GO TO 490
  480 CONTINUE
      KK=KK+3
      SCREEN(KK)=FLOAT(NTBL(N)-NPREVT)-0.85*FLOAT(NPREVS)-0.35*FLOAT
     1 (MAXO(NSPDF(KK)-1,0))
      NPREVT=NPREVT+NPREVS
      NPREVS=NSPDF(KK)
      IF (KK.LT.K3) GO TO 470
C
C     CALCULATE THE ENERGY OF THIS ION STATE
C
  490 K4=KC+1-K2
      ENIK(N,K1,K4)=0.
      DO 500 K=1,K3
  500 ENIK(N,K1,K4)=ENIK(N,K1,K4)+SCREEN(K)**2*FLOAT(NSPDF(K))/XNSTAR(K)
      ENIK(N,K1,K4)=1.356E-2*ENIK(N,K1,K4)
C
C     DIVIDE THE VALUE OF THE SCREENING CONSTANT ASSOCIATED WITH THIS
C        SUB-SHELL BY THE VALUE OF THIS SAME CONSTANT FOR THE UNIONIZED
C        ATOM AND STORE AS SNIK(ELEMENT,IONIZATIONLEVEL,SUB-SHELL) --
C        ALSO STORE THE VALUE OF THE NUMBER OF ELECTRONS IN THE
C        SUB-SHELL GROUPS -- S, P, D, AND F -- AS NNIK(N,K1,K)
C
      SNIK(N,K1,K4)=SCREEN(K2)
      IF (NSTMP.LE.0) GO TO 530
      IF (K2-1) 510,510,520
  510 SNIK(N,K1,K4)=SNIK(N,K1,K4)-0.3
      GO TO 530
  520 SNIK(N,K1,K4)=SNIK(N,K1,K4)-0.35
  530 SNIK(N,K1,K4)=SNIK(N,K1,K4)/SCRENO(N,K4)
  540 NSPDF(K2)=NSTMP+1
      DO 550 K=1,KI
      KL=NOEC(N)+1-K
  550 NNIK(N,K1,KL)=NGRUP(K)
      DO 560 K=1,14
  560 NSNIK(N,K1,K)=NSPDF(K)
      GO TO 600
C
C     RESET FOR NI = 0
C
  570 DO 580 K=1,14
  580 SNIK(N,1,K)=1.
      DO 590 K=1,19
  590 NNIK(N,1,K)=NION(N,K)
      IF (NVARE(N).EQ.1.OR.ZF(N,J).LE.0.5) GO TO 610
  600 CONTINUE
      IF (J.EQ.JTS.AND.JPRIN.EQ.1) WRITE (6,1230) N,((ENIK(N,K1,K),K=1,1
     1 4),K1=1,8)
  610 CONTINUE
      GO TO 650
C
C     WHEN NVARM(MATERIAL) IS NOT EQUAL TO ZERO, THE X-SECTIONS OF ALL
C        THE ELEMENTS ARE FIXED AT THE COLD VALUE -- SOME CONSTANTS
C        MUST BE SET
C
      ENTRY OPAGC
  620 CONTINUE
      DO 640 N1=1,NEM
      N=IELEM(M,N1)
      DO 630 K=1,14
  630 SNIK(N,1,K)=1.
      DO 640 K=1,19
  640 NNIK(N,1,K)=NION(N,K)
C
C     CALCULATE THE ENERGY DEPOSITION IN THIS ZONE
C
  650 DO 1220 NS=1,NSPEC
      IF (J.EQ.JTS.AND.JPRIN.EQ.1) WRITE (6,1240) ((NNIK(1,N1,K),K=1,19)
     1 ,N1=1,8)
      IF (TIME-START(NS)) 1220,1220,660
  660 IF (TIME-SSTOP(NS)) 670,670,1220
  670 NNU=NHNU(NS)
      NBBS=NBB(NS)
      SS(J,NS)=0.
      ESUM=0.
      IF (J-2) 680,680,750
  680 CONTINUE
*IF DEF,B64
      SDURM=AMIN1(SDURM,SSTOP(NS)-START(NS))
*ENDIF
```

```
*IF DEF,B32
      SDURM=DMIN1(SDURM,SSTOP(NS)-START(NS))
*ENDIF
      IF (NNU) 690,710,690
  690 DO 700 I=1,NNU
  700 ESJ(1,I)=ES(1,I)
      GO TO 760
  710 DO 740 L=1,NBBS
      DO 740 I=1,109
      IF (I-99) 720,720,730
  720 ESJ(L,I)=EE(L)*ANGLE*0.01
      GO TO 740
  730 ESJ(L,I)=EE(L)*ANGLE*0.001
  740 CONTINUE
  750 IF (NNU.EQ.0) NNU=109
C
  760 DO 1210 L=1,NBBS
      IF (JPRIN.NE.1.OR.J.NE.JTS.OR.NKEEP/NPRIN*NPRIN.NE.NKEEP) GO TO 77
     1 0
      WRITE (6,1300) NS,L,TEMP(J)
  770 DO 780 N1=1,NEM
      N=IELEM(M,N1)
      KMAX(N)=MIN0(NTBL(N),8)
      KN=KMAX(N)
      DO 780 K1=1,KN
  780 ISTART(N,K1)=0
      DO 1200 I=1,NNU
      IF (ESJ(L,I)-1.E-20) 1200,1200,790
  790 ACION=0.
      IPRIN=0
      IF (KPRIN.NE.0.AND.J.EQ.JTS.AND.I/10*10.EQ.I) IPRIN=1
      ZBAR=0.0
      DO 1170 N1=1,NEM
C     N IS ELEMENT INDEX
      N=IELEM(M,N1)
C     KN IS NUMBER OF CALCULATED EDGES (NUMBER OF OCCUPIED ELECTRON
C       SUB-SHELLS-1S,2S,2P,3S,3P,3D,4S ETC)
      KN=NOEC(N)
C     KNI IS KN WITH S+P SUBSHELLS CONDENSED INTO ONE GROUP FOR EACH N
      KNI=ITABL(KN)
C     KGO IS THE NUMBER OF DEGREES OF IONIZATION (BEFORE XRAY EVENT)
C       TO INCLUDE CONTRIBUTIONS TO X SEC
      KGO=KMAX(N)
      IF (J-JHAT) 800,800,840
  800 IF (NVARM(M)) 840,810,840
  810 IF (NVARE(N)) 840,820,840
  820 IF (TEMP(J)-1.2E4) 840,840,830
  830 IF (ZF(N,J)-0.5) 840,840,850
C     ATOM IS COLD-XSEC IS CONSTANT
  840 KGO=1
```

```
      JI=C
      NOEI(N.1)=KN
  850 RHON=XAW(N)*FLOAT(NATOM(M))*AF(M,N)/XMW(M)/V(J)
      IF (IPRIN.EQ.0) GO TO 860
      WRITE (6,1280) J,N,TIME
      WRITE (6,1310) I,NAMEL(N),ZF(N,J),KGO
      WRITE (6,1250) KN,KNI,KGO,KMAX(N),(NOEI(N,IJ),IJ=1,8),(NLEC(N,IJ)
     1 ,IJ=1,6),(NI(IJ,N),IJ=1,8)
C
C     CALCULATE CONTRIBUTIONS TO XSEC FROM THE VARIOUS
C      DEGREES OF IONIZATION
C
  860 DO 1160 K1=1,KGO
C
C     INITIALIZE THE ABSORPTION EDGE FOR THIS STATE OF THE ELEMENT TO
C        THE LOWEST POSSIBLE AND FIND HIGHEST ACTIVE EDGE
C
C     KI IS HIGHEST OCCUPIED SUBSHELL
      KI=NOEI(N,K1)
      IF (KGO.EQ.1) GO TO 880
C     NUMBER OF ELECTRONS PREVIOUSLY REMOVED
      JI=NI(K1,N)
C     IF ALL ELECTRONS ARE GONE, THERE IS NO INTERACTION
      IF (JI.GE.NTBL(N)) GO TO 1160
      DO 870 NTMP=1,15
      NTMP1=16-NTMP
      IF (KI.LE.NLEC(N,NTMP1)) GO TO 890
  870 CONTINUE
  880 NTMP=0
      GO TO 900
C     INDEX NTMP USED FOR BOUND-BOUND TRANSITIONS
  890 NTMP=NTMP1
  900 IF (ISTART(N,K1)) 910,910,970
  910 NEDGLN(N,K1,L)=KN+1-KI
C     K2 IS LOWEST ACTIVE EDGE-CORRESPONDS TO HIGHEST OCC SHELL KI
      K2=NEDGLN(N,K1,L)
      K3=KN+1-K2
      K3=MAX0(K3,1)
C     K3 IS EDGE NUMBER K2 WITH WITH S+P SHELLS CONDENSED
C     AND COUNTS IN REVERSED ORDER OF ENERGY DECREASING
C     THE ELECTRON TO BE REMOVED FROM SUB-SHELL CORRESPONDING TO K3
      K3=KNI+1-ITABL(K3)
      IF (IPRIN.EQ.1) WRITE (6,1260) N,K1,L,K2,K3
      IF (KGO.EQ.1.OR.JI.EQ.0) GO TO 920
C     SHIFT IS I(N,J,K) - I(N,0,K) IN MANUAL
C     EION(N,JI) IS ELECTRON BINDING ENERGY REMAINING IN ION WITH
C       JI ELECTRONS REMOVED (PRIOR TO XRAY INTERACTON)
C     ENIK(N,K1,K3) IS BINDING ENERGY LEFT AFTER REMOVING AN
C       ELECTRON FROM SHELL K3, FROM THE ABOVE ION
C     EDGEC(N,K2) IS CALCULATED EDGE K2
```

```
      SHIFT(N,K1,K3)=EION(N,JI)-ENIK(N,K1,K3)-EDGEC(N,K2)
      GO TO 930
  920 SHIFT(N,1,K3)=0.0
  930 IF (NTMP.NE.0.AND.JI.NE.0) GO TO 940
      RBB=1.0
      GO TO 950
C     RBB IS ENERGY FRACTION FOR BOUND-BOUND TRANSITIONS
  940 RBB=EBB(N,NTMP,K3)
      RBB=RBB/(EION(N,JI)-ENIK(N,K1,K3))
      IF (RBB.LE.0.0) RBB=1.0
C     SKIP CALCULATION IF LOWEST EDGE IS GREATER THAN PHOTON ENERGY
  950 IF (TBL(I)*T(L)-RBB*(EDGE(N,K2)+SHIFT(N,K1,K3))) 1160,1160,960
  960 ISTART(N,K1)=1
      GO TO 980
  970 K2=NEDGLN(N,K1,L)
  980 K4=KN-K2
      K4=MAX0(K4,1)
C     K4 IS EDGE NUMBER K2 WITH WITH S+P SHELLS CONDENSED
C     COUNTING IN REVERSED ENERGY ORDER
C     THE ELECTRON TO BE REMOVED FROM SUB-SHELL CORRESPONDING TO K4
      K4=KNI+1-ITABL(K4)
      IF (IPRIN.EQ.1) WRITE (6,1270) K2,K4
      IF (KGO.EQ.1.OR.JI.EQ.0) GO TO 990
C     CHANGED FOLLOWING EDGEC INDEX FROM K2+1
      SHIFT(N,K1,K4)=EION(N,JI)-ENIK(N,K1,K4)-EDGEC(N,MIN0(K2+1,KN))
      GO TO 1000
  990 SHIFT(N,1,K4)=0.0
 1000 IF (NTMP.NE.0.AND.JI.NE.0) GO TO 1010
      RBB=1.0
      GO TO 1030
 1010 RBE=EBB(N,NTMP,K4)
      KNN=KC+1-K4
      IF (NSNIY(N,K1,KNN).EQ.0) GO TO 1020
      RBB=RBB/(EION(N,JI)-ENIK(N,K1,K4))
 1020 CONTINUE
      IF (RBB.LE.0.0) RBB=1.0
C     CHECK IF EDGE K2+1 IS LESS THAN PHOTON ENERGY
 1030 IF (TBL(I)*T(L)-RBB*(EDGE(N,K2+1)+SHIFT(N,K1,K4))) 1050,1050,1040
C     INCREMENT K2 TO FIND HIGHEST ACTIVE EDGE
 1040 K2=K2+1
C
C . . IF K2 .GT. KN, RESET TO KN AND BREAK OUT OF LOOP
C
      IF (K2.LE.KN) GO TO 980
      K2=MIN0(K2,KN)
 1050 NEDGLN(N,K1,L)=K2
      IF (IPRIN.EQ.0) GO TO 1070
      WRITE (6,1320) K1,K2,K4,NTMP,RBB
      IF (RBB.EQ.1) GO TO 1060
      IF (EDGEC(N,K2+1).EQ.0.0) EDGEC(N,K2+1)=EDGE(N,K2+1)
      WRITE (6,1330) EION(N,JI),ENIK(N,K1,K4),EDGEC(N,K2+1)
 1060 CONTINUE
      WRITE (6,1340)
C
C        TBL(I)*T(L).....
C
 1070 KK=KN+1-KI
      IF (KK-K2) 1080,1080,1160
C     LOOP OVER ALL ACTIVE EDGES K (AN ELECTRON REMOVED FROM
C       CORRESPONDING SUB-SHELLS
 1080 DO 1150 K=KK,K2
      IF (AA(N,K)) 1150,1150,1090
 1090 KJ=MINO(K,KN)
      K3=KN+1-KJ
      K3=MAXO(K3,1)
      K3=KNI+1-ITABL(K3)
C     POPULATION OF IONS WITH DEGREE JI
      POP=R(K1,N)
      JI=NI(K1,N)
      IF (KGO.NE.1.AND.JI.NE.0) GO TO 1100
      IF (KGO.EQ.1) POP=1.0
      JI=0
      SHIFT(N,1,K3)=0.0
      GO TO 1110
 1100 SHIFT(N,K1,K3)=EION(N,JI)-ENIK(N,K1,K3)-EDGEC(N,KJ)
C     NION IS NUMBER OF ELECTRONS IN EACH SUB-SHELL IN COLD ATOM
 1110 IF (JI.GE.NTBL(N).OR.NION(N,KJ).EQ.0) GO TO 1150
      IF (TBL(I)*T(L)-SHIFT(N,K1,K3)) 1120,1120,1130
 1120 ETMP=TBL(1)*T(L)/2.0
      GO TO 1140
 1130 ETMP=TBL(I)*T(L)-SHIFT(N,K1,K3)
C     NNIK IS NUMBER OF ELECTRONS IN EACH SUBSHELL K IN ION PRIOR TO
C       X-RAY
 1140 ACION=ACION-RHON*POP*AA(N,K)*ETMP**(B(N,K)+1.)*FLOAT(NNIK(N,K1,KJ)
     1 )/FLOAT(NION(N,KJ))*SNIK(N,K1,K3)**5/ANGLE
      IF (IPRIN.EQ.0) GO TO 1150
      WRITE (6.1350) K,K3,RHON,POP,JI,NNIK(N,K1,KJ),NION(N,KJ),SHIFT(N
     1 ,K1,K3) SNIK(N,K1,K3),AA(N,K),B(N,K),ACION,ETMP
      IF (SHIFT(N,K1,K3).EQ.0.) GO TO 1150
      WRITE (6,1360) EION(N,JI),ENIK(N,K1,K3),EDGEC(N,KJ),SCRENO(N,K3)
 1150 CONTINUE
 1160 CONTINUE
 1170 ZBAR=ZBAR+AF(M,N)*ZF(N,J)**2
      ACION=ACION/TBL(I)/T(L)
      IF (IPRIN.EQ.1) WRITE (6,1370) I,ACION
      IF (TEMP(J).LT.1.2E4) GO TO 1180
      ACION=ACION-9.9E3*(FLOAT(NATOM(M))/XMW(M))**2*ZBAR*ZFM(J)/(TBL(I)
     1 *T(L))**3/TEMP(J)**0.5/V(J)**2
C
C     CALCULATE THE ENERGY DEPOSITION IN THIS ZONE FROM THIS SOURCE
```

```
C
 1130 AX1=ACION*(X(J)-X(J-1))
      IF (J.EQ.2.OR.J.EQ.20.OR.J.EQ.54) AX2=-ACION/RHON
      IF (J.EQ.2.OR.J.EQ.20.OR.J.EQ.54) WRITE (7,1290) J,TIME,AX2
      IF (AX1.LT.-20.) EIZ=ESJ(L,I)
      IF (AX1.LT.-20.) GO TO 1190
      EIZ=ESJ(L,I)*(1.-EXP(ACION*(X(J)-X(J-1))))
 1190 ESJ(L,I)=ESJ(L,I)-EIZ
      ESUM=ESUM+EIZ
C     STORE X-RAY FLUX IN XFX(J)
      XFX(J)=XFX(J)+ESJ(L,I)/(SSTOP(NS)-START(NS))
 1200 CONTINUE
 1210 CONTINUE
C
C     CALCULATE THE ENERGY DEPOSITION RATE IN THIS ZONE FOR EACH
C        SPECTRUM IN UNITS OF ERGS/GRAM/SECOND
C
      SS(J,NS)=ESUM*4.186E7*V(J)/(X(J)-X(J-1))/(SSTOP(NS)-START(NS))
 1220 CONTINUE
      RETURN
C
 1230 FORMAT ('OENIK',I5/(1X,1P10E12.4/1X,4E12.4))
 1240 FORMAT ('ONNIK=',19I6)
 1250 FORMAT ('OKN,KNI,KGO,KMAX-NOEI-NLEC-NI',4I8/1X,8I8/1X,6I8/1X,8I8)
 1260 FORMAT ('ONEDGLN(',3I5,')=',2I8)
 1270 FORMAT ('OK2,K4 SET=',2I8)
 1280 FORMAT (/5X,20HFLOION CALC FOR ZONE,I3,7H CYCLE ,I4,6H TIME=,1PE10
     1 .3)
 1290 FORMAT (I5,2(1PE15.8,2X))
 1300 FORMAT (///31H CALCULATIONS FOR ZONE 2,NSPEC=,I2,20X,4HNBB=,I2,20X
     1 ,5HTEMP=,E11.3)
 1310 FORMAT (/17H ENERGY INTERVAL=,I3,3X,11HELEMENT IS ,A10,3X,14HIONIZ
     1ATION IS ,E11.3,3X,27HNUMBER OF ACTIVE LEVELS IS ,I2)
 1320 FORMAT (14H LEVEL NUMBER=,I2,20X,32HHIGHEST NUMBERED ACTIVE EDGE I
     1S ,I2/4H K4=,I2,10X,5HNTMP=,I2,10X.4HRBB=,E11.3)
 1330 FORMAT (6H EION=,1PE13.5,10X,5HENIK=,E13.5,10X,10HNEXT EDGE=,E11.3
     1 )
 1340 FORMAT (1X.4HEDGE,6X,7HS+P,D,F.5X,3HRHO,9X,3HPOP.5X,2HJI.1X,4HNNIK
     1 ,1X,4HNION,3X,5HSHIFT,8X,4HSNIK,9X.2HAA,10X.1HB,10X,2HMU,8X,4HETM
     2P)
 1350 FORMAT (I4,I7,E18.6,E12.6,I4,I4,I5,E12.6,E12.6,E12.6,E12.6,E12.6
     1 ,E12.6)
 1360 FORMAT (16H ENERGY OF ATOM=,E12.6,2X,29HENERGY WITH ELECTRON REMOV
     1ED=,E12.6,2X,6HEDGEC=,E12.6,2X.15HATOM SCREENING=,E12.6)
 1370 FORMAT (34H CROSS-SECTION FOR ENERGY INTERVAL,I3,3HIS ,E11.3)
      END
```

```
*DECK PE
      SUBROUTINE PE
*IF DEF,B32
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
*ENDIF
*CALL BLANK
*CALL EQVP
*CALL EQFL
*CALL INDX
*CALL PEPT
      ENU=1./V(J)/RHO(M)
      IF (ENU.LT.1.) GO TO 30
      ES1=0.0
C
C     MATERIAL IS COMPRESSED.
C
      EMU=ENU-1.
      IF (CUSPA(M).EQ.0.) GO TO 10
      ARG=EMU-CUSPA(M)
      IF (ARG.LE.0.) GO TO 10
      PH=(CUSP1(M)+((CUSPS(M)*ARG+CUSPD(M))*ARG+CUSPC(M))*ARG)
      GO TO 20
   10 PH=((EQSTS(M)*EMU+EQSTD(M))*EMU+EQSTC(M))*EMU
   20 CONTINUE
      ES10=FLOAT(NATOM(M))/XMW(M)
      DEDT=1.25E8*(2.+3.*ZFM(J))*ES10
      DPDT=8.31E7*(3.*EQSTG(M)+2.*ZFM(J))*ES10/V(J)
      TREF=300.
      PH=PH*(1.-.5*EQSTG(M)*EMU/ENU)
      EH=-.5*PH*(V(J)-1./RHO(M))
      ENTRY PE1
      DFDT=(TEMPJ-TREF)*ES10
      ET(J)=EI(J)+2.5E8*(1.+.5*ZFM(J))*TEMPJ*ES10+EH
      PN(J)=PH+8.31E7*(3.*EQSTG(M)+ZFM(J))*DFDT/V(J)
      IGO=1
      GO TO 80
C
C     MATERIAL IS EXPANDED
C
   30 CONTINUE
      V1=1./ENU
      ENU2=EQSTN(M)*(1.-V1)*V1
      IF (ENU2.LE.-10.) GO TO 40
      ES1=EQSTE(M)*(1.-EXP(ENU2))
      GO TO 50
   40 ES1=EQSTE(M)
   50 ALF=EQSTH(M)+(EQSTG(M)-EQSTH(M))*SQRT(ENU)
      IF (ABS(ALF-EQSTH(M)).GT.1.E-3) GO TO 60
      W1=1.0
      GO TO 70
   60 W1=2.*(EQSTG(M)-ALF)/(ALF-EQSTH(M))
      W1=(2.+W1)/(1.+W1)
   70 CONTINUE
      ES10=FLOAT(NATOM(M))/XMW(M)
      DEDT=1.25E8*(W1+3.*ZFM(J))*ES10
      DPDT=8.31E7*(1.5*ALF*W1+2.*ZFM(J))*ES10/V(J)
      TREF=ES1/(1.25E8*ES10*(W1+.5*ZFM(J)))
*IF DEF,B64
      TREF=AMAX1(TREF,300.)
*ENDIF
*IF DEF,B32
      TREF=DMAX1(TREF,3.D2)
*ENDIF
      ENTRY PE2
      DFDT=(TEMPJ-TREF)*ES10
      ET(J)=EI(J)+1.25E8*ES10*TEMPJ*(W1+ZFM(J))
      PN(J)=8.31E7*(1.5*ALF*W1+ZFM(J))*DFDT/V(J)
      IGO=2
   80 CONTINUE
      RETURN
C
      END
```

```
*DECK PLOT
      SUBROUTINE PLOT
*IF DEF,B32
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
*ENDIF
C
*CALL BLANK
*CALL PLOTCM
      DATA IFEDIT, IJEDIT /1,0/
C
      WRITE (2) (DISCPT(I),I=1,8)
      DO 10 I=1,NMTRLS
      JBND(I)=25*I+1
   10 CONTINUE
      JBND(NMTRLS)=JSTAR-1
      PPMAX=0.
      PPMIN=0.
      WRITE (6,80)
      WRITE (6,90)
      DO 20 J=1,JSTAR
      PSMAX(J)=PSMAX(J)*1.E-9
      PSMIN(J)=PSMIN(J)*1.E-9
*IF DEF,B64
      PPMAX=AMAX1(PSMAX(J),PPMAX)
      PPMIN=AMIN1(PSMIN(J),PPMIN)
*ENDIF
*IF DEF,B32
      PPMAX=DMAX1(PSMAX(J),PPMAX)
      PPMIN=DMIN1(PSMIN(J),PPMIN)
*ENDIF
      WRITE (6,100) PX(J),PSMAX(J),PSMIN(J)
   20 CONTINUE
      JSTAR=JSTAR-1
      WRITE (2) JSTAR,NMTRLS,(JBND(I),I=1,NMTRLS),PPMIN,PPMAX
      WRITE (2) (PX(J),PSMAX(J),PSMIN(J),J=1,JSTAR)
      IF (NJEDIT.GT.0) GO TO 30
      M=0
      NJEDIT=1
      WRITE (2) M,IJEDIT,NJEDIT,MTLN(1),DSTF(1),JEDIT(1),MM(1)
      RETURN
   30 CONTINUE
      REWIND 8
      DO 40 I=1,NJEDIT
      JJ=MTLN(I)
      MM(I)=MATL(JJ)
   40 CONTINUE
      N=N-1
      IF (IFEDIT.GT.C) WRITE (6,110)
      WRITE (2) N,IJEDIT,NJEDIT,(MTLN(I),DSTF(I),JEDIT(I),MM(I),I=1
     1 ,NJEDIT)
      AB=1.E-9
      IF (IJEDIT.GT.0) AB=1.E-6
      DO 70 L=1,N
      READ (8,120,END=50) TIME,(SQJ(I),I=1,NJEDIT)
   50 TIME=TIME*1.E+6
      DO 60 I=1,NJEDIT
      SQJ(I)=SQJ(I)*AB
   60 CONTINUE
      WRITE (2) TIME,(SQJ(LL),LL=1,NJEDIT)
      IF (IFEDIT.GT.0) WRITE (6,120) TIME,(SQJ(I),I=1,NJEDIT)
   70 CONTINUE
      REWIND 2
      RETURN
C
   80 FORMAT (1H1,//,10X,37HPEAK COMPRESSIVE AND TENSILE ENVELOPE/)
   90 FORMAT (12X,5HX(CM),6X,8HSMAX(KB),4X,8HSMIN(KB),/)
  100 FORMAT (9X,3(1PE10.3,2X))
  110 FORMAT (1H1,10X,11HJEDIT PRINT,/)
  120 FORMAT (BZ,3X,1P11E12.4/)
      END
```

```
*DECK PT
      SUBROUTINE PT
*IF DEF,B32
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
*ENDIF
*CALL BLANK
*CALL EQVP
*CALL EQFL
*CALL INDX
*CALL PEPT
      M=1
      DO 170 J=2,J1
      IF (J-1.EQ.JBND(M)) M=M+1
      NTMP=NELEM(M)-1
      NTMP=MAX0(NTMP,1)
      ZFM(J)=ZFM(J)/FLOAT(NATOM(M))
      IF (IFLOW.EQ.0.AND.ITER(J).EQ.0) GO TO 160
      IF (ITER(J-1).EQ.0.AND.ITER(J).EQ.0.AND.ITER(J+1).EQ.0) GO TO 160
      TEMPJ=TEMP(J)
      ICOUNT=1
      TMIN=0.
      TMAX=1.E40
      FTNE1=-E(J)-EADD(J)+0.5*P(J)*DV(J)-0.5*(FO(J)-FO(J-1))*(DTN-DTNH)
     1 /ZM(J)
      ITER(J)=4
      IF (IDIF.EQ.0) ITER(J)=1
      IF (ICON.EQ.0) ITER(J)=2
      IF (ICON.EQ.0.AND.IDIF.EQ.0) ITER(J)=3
      CALL PE
      GO TO 30
   10 CALL PE1
      GO TO 30
   20 CALL PE2
   30 CONTINUE
C
C     FTNEW IS THE ENERGY IMBALANCE CAUSED BY USE OF INCORRECT TEMP.
C
      FTNEW=FTNE1+ET(J)+0.5*PN(J)*DV(J)
      IF (IFLOW.EQ.0) GO TO 70
      FTMP=0.0
      F(J)=0.0
      IF (J.EQ.JFIN) GO TO 60
      IF (ITER(J).EQ.1) GO TO 40
      F(J)=2.0*XCON(M)/(X(J+1)-X(J-1))*(TEMP(J+1)-TEMPJ)
   40 IF (ZFM(J).LT.0.01.AND.ZFM(J+1).LT.0.01) GO TO 50
      XLTP1=XLAM1(J)*(TEMP(J)/TEMPJ)**0.5
      ARGEXP=TEMPJ/TEMP(J)
      ARGTST=10.**(100./(3.5*FLOAT(NTMP)))
*IF DEF,B64
```

```
      ARGEXP=AMIN1(ARGEXP,ARGTST)
*ENDIF
*IF DEF,B32
      ARGEXP=DMIN1(ARGEXP,ARGTST)
*ENDIF
      XLTP2=XLAM2(J)*ARGEXP**(3.5*FLOAT(NTMP))
      IF (XLTP2/(X(J)-X(J-1)).GT.5.67E-5) XLTP2=5.67E-5*(X(J)-X(J-1))
      XLTP3=XLAM2(J+1)
      IF (XLTP3/(X(J+1)-X(J)).GT.5.67E-5) XLTP3=5.67E-5*(X(J+1)-X(J))
      FTMP=(XLAM1(J+1)+XLTP3+XLTP1+XLTP2)/(X(J+1)-X(J-1))*(TEMP(J+1)**4
     1 -TEMPJ**4)
      F(J)=F(J)+FTMP
   50 IF (J.NE.2) GO TO 70
   60 FTMP1=5.67E-5*TEMPJ**4
      IF (J.EQ.2) F(J-1)=FTMP1
      IF (J.EQ.JFIN) F(J)=-FTMP1
   70 FTMP=F(J)-F(J-1)
C
C     TNEW IS THE NEW GUESS AT A TEMPERATURE
C
      FTNEW=FTNEW-0.5*FTMP*DTNH/ZM(J)
      IF (ICOUNT.NE.1) GO TO 80
      SAVE1=ET(J)
      SAVE2=PN(J)
      SAVE3=FTNEW
      FSAVE1=F(J)
      FSAVE2=F(J-1)
   80 TNEW=TEMPJ-FTNEW/(DEDT+0.5*DV(J)*DPDT)
      IF (KPRIN.NE.0.AND.J.EQ.JTS) WRITE (6,180) J,FTNE1,ET(J),DV(J),PN
     1 (J),DEDT,DPDT,DFDT,TEMPJ,TNEW,FTNEW,XLAM1(J),XLAM2(J),F(J),F(J-1)
     2 ,FO(J),FO(J-1),DTNH,ZM(J),TMIN,TMAX
*IF DEF,B64
      IF (ABS(TNEW-TEMPJ).LE.AMAX1(DTMIN*(TNEW-300.0)/2.0,1.E-7)) GO TO
     1 120
*ENDIF
*IF DEF,B32
      IF (ABS(TNEW-TEMPJ).LE.DMAX1(DTMIN*(TNEW-300.0)/2.0,1.D-7)) GO TO
     1 120
*ENDIF
      IF (TNEW.LT.TEMPJ) GO TO 90
      IF (TEMPJ.GT.TMIN) TMIN=TEMPJ
      IF (TNEW.GT.TMAX) TNEW=TMAX
      GO TO 100
   90 IF (TEMPJ.LT.TMAX) TMAX=TEMPJ
      IF (TNEW.LT.TMIN) TNEW=TMIN
  100 TEMPJ=0.5*(TEMPJ+TNEW)
      ICOUNT=ICOUNT+1
      IF (ICOUNT.LE.500) GO TO 110
      WRITE (6,190) J
      JTS=J
      WRITE (6,180) J,FTNE1,ET(J),DV(J),PN(J),DEDT,DPDT,DFDT,TEMPJ,TNEW
     1 .FTNEW,XLAM1(J),XLAM2(J),F(J),F(J-1),FO(J),FO(J-1),DTNH,ZM(J)
     2 .TMIN,TMAX
      JCYCS=MIN0(N+5,JCYCS)
      GO TO 160
  110 CONTINUE
      GO TO (10,20), IGO
      CALL GOTOER
  120 TEMPO(J)=TEMP(J)
      IF (TEMPJ.GT.1.2E4.OR.EI(J).GT.1.E-10) GO TO 130
      ITER(J)=0
      TEMP(J)=TEMPJ
      ET(J)=ET(J)-FTNEW
      GO TO 160
  130 IF (ABS(TEMPJ-TEMP(J))-DTMIN*(TEMP(J)-300.0)) 140,140,150
  140 ITER(J)=0
      ET(J)=SAVE1-SAVE3
      PN(J)=SAVE2
      IF (IFLOW.EQ.0) GO TO 160
      F(J)=FSAVE1
      F(J-1)=FSAVE2
      GO TO 160
  150 TEMP(J)=0.9*TEMP(J)+0.1*TNEW
      IF (TEMP(J).GT.1.20*TEMPO(J)) TEMP(J)=1.20*TEMPO(J)
      IF (TEMP(J).LT.TEMPO(J)/1.20) TEMP(J)=TEMPO(J)/1.20
  160 IF (ITER(J).EQ.0) JCOUNT=JCOUNT+1
      ZFM(J)=ZFM(J)*FLOAT(NATOM(M))
  170 CONTINUE
      RETURN
C
  180 FORMAT (6X,1HJ,7X,5HFTNE1,10X,2HET,10X,2HDT,10X,2HPN,8X,4HDEDT,8X,
     1 4HDPDT,8X,4HDFDT,7X,5HTEMPJ,8X,4HTNEW/4X,I3,1P9E12.4//14X,5HFTNEW
     2 ,7X,5HXLAM1,7X,5HXLAM2,8X,4HF(J),6X,6HF(J-1),7X,5HFO(J),5X,7HFO(J
     3-1),8X,4HDTNH,10X,2HZM/7X,1P9E12.4//15X,4HTMIN,8X,4HTMAX/7X,1P2E12
     4 .4///)
  190 FORMAT (10H J EQUALS ,I10,5X,17HHAS NOT CONVERGED)
      END
```

```
*DECK REZONE
      SUBROUTINE REZONE
*IF DEF,B32
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
*ENDIF
C
C         HYPUF  REZONING  ROUTINE
C
*CALL BLANK
*CALL EQVP
*CALL PLOTCM
*CALL RZCOM
*CALL SPLLC
      DIMENSION ZFR(10), ZFL(10), SSR(10), SSL(10)
      DATA JDBG /0/
C
C          LOOP OVER MESH
C
      WTAPE=0.0
      IF (JDBG.EQ.1) CALL EDIT
      JJPRIN=3
      M=1
      J=2
      JJPRIN=J
      SMAX=ABS(SMAX)
   10 AAA=ABS(S(J)-RZC1)
      IF (JDBG.EQ.1) WRITE (6,660) JFIN
      AAA=AAA/SMAX
      IF (J-1.EQ.JBND(M)) M=M+1
      IF (J.GE.200) GO TO 600
      IF (J.GE.JFIN) GO TO 600
      ARZCO=AAA*RZCO
*IF DEF,B64
      SLIM=AMAX1(RSCRIT,ARZCO)
*ENDIF
*IF DEF,B32
      SLIM=DMAX1(RSCRIT,ARZCO)
*ENDIF
      DELS=(S(J+1)-S(J))/SMAX
      ADELS=ABS(DELS)
      IF (JFIN.GE.200) GO TO 20
      IF (J.GT.JSTAR) GO TO 600
C
      JF=0
      IF (JDBG.EQ.1) WRITE (6,670) J
      IF (JDBG.EQ.1) WRITE (6,680) AAA,ARZCO,SLIM,DELS,ADELS,S(J),SMAX
C
C         CHECK FOR DIVIDE
C
```

```
      IF (ADELS.GT.SLIM) GO TO 30
C
C        CHECK FOR COMBINE
C
   20 CONTINUE
      IF (ADELS.LT.(0.2*SLIM)) GO TO 410
      J=J+1
      IF (J.LT.JFIN) GO TO 10
      GO TO 600
C
C        DIVIDE  LOOP
C
   30 AU=ABS(U(J))
      DISTRZ=.5*DTNH*CS(J)*NREZON*U(J)/AU
      IF (JDBG.EQ.1) WRITE (6,690) DISTRZ,DTNH,CS(J),NREZON,U(J),AU
      IF (DISTRZ.GT.0.0) GO TO 60
      DO 40 LJ=1,J
      LM1=LJ-1
      DELX=X(LJ)-X(J)
      IF (DELX.LT.DISTRZ) GO TO 50
   40 CONTINUE
   50 JF=MAX0(1,LM1)
      GO TO 250
   60 DO 70 LJ=J,JFIN
      DELX=ABS(X(LJ)-X(J))
      IF (DELX.GE.DISTRZ) GO TO 80
      IF (JDBG.EQ.1) WRITE (6,700) J,LJ,JFIN,DELX
   70 CONTINUE
C
   80 NRZ1=LJ-J
      NRZ2=NRZ1*(1.1**NRZ1)
      JF=MIN0(J+NRZ2,JFIN)
C
C       REZONE TO RIGHT ...
C
      J1=J
      DO 240 LJ=J1,JF,2
      LM1=LJ-1
      LP1=LJ+1
      IF (JDBG.EQ.1) WRITE (6,710) J,JF,LJ
      ZMT=2.0*ZM(LJ)
      IF (ZM(LM1).GE.ZMT) GO TO 230
      IF (ZM(LP1).GE.ZMT) GO TO 230
      XL=(X(LJ)+X(LM1))/2.0
      XR=X(LJ)
      DELX=X(LJ)-X(LM1)
      XMIN=1.E-8*CS(LJ)
      IF (DELX.LE.XMIN) GO TO 230
C
C         CHECK FOR MATERIAL BOUNDRIES
```

```
C
      DO 90 LM=1,NMTRLS
      IF (LJ.EQ.JBND(LM)) GO TO 110
      IF ((LM1).EQ.JBND(LM)) GO TO 100
   90 CONTINUE
      IF (TSPALL(LJ).EQ.1.234) GO TO 240
      IF (TSPALL(LJ).EQ.0.) GO TO 100
      IF (TSPALL(LM1).EQ.1.234) GO TO 100
      IF (TSPALL(LM1).NE.TSPALL(LJ)) GO TO 100
      IF (TSPALL(LJ).NE.TSPALL(LP1)) GO TO 110
C
C     DIVIDE INTERNAL ZONE
C
      RZRL=(ZM(LM1)+ZM(LJ))/(ZM(LM1)+ZM(LJ)+ZM(LP1))
      RZRR=1.0-RZRL
      L1=LP1
      L2=LP1
      L3=LM1
      L4=LM1
      GO TO 120
C
C       DIVIDE ZONE WITH MATERIAL BOUNDRY AT LEFT INTERFACE ...
C
  100 CONTINUE
      RZRR=(0.5*ZM(LJ)+ZM(LP1))/(ZM(LJ)+ZM(LP1))
      RZRL=1.-RZRR
      L1=LP1
      L2=LP1
      L3=LJ
      L4=LP1
      GO TO 120
C
C        DIVIDE ZONE WITH MATERIAL BOUNDRY AT RIGHT INTERFACE ...
C
  110 CONTINUE
      RZRL=(ZM(LM1)+0.5*ZM(LJ))/(ZM(LM1)+ZM(LJ))
      RZRR=1.-RZRL
      L1=LJ
      L2=LM1
      L3=LM1
      L4=LM1
  120 CONTINUE
      ZMR=0.5*ZM(LJ)
      ZML=ZMR
      UR=U(LJ)
      UL=(ZMR*U(LM1)+ZML*U(LJ))/ZM(LJ)
C
      IF (JDBG.EQ.1) WRITE (6,720) DR,DL,V(LM1),V(LJ),V(LP1),ZMR,ZML,ZM
     1 (LJ)
      SR=S(L1)+RZRR*(S(LJ)-S(L2))
      SL=S(L3)+RZRL*(S(LJ)-S(L4))
      SDR=SD(L1)+RZRR*(SD(LJ)-SD(L2))
      SDL=SD(L3)+RZRL*(SD(LJ)-SD(L4))
      SD2R=SD2(L1)+RZRR*(SD2(LJ)-SD2(L2))
      SD2L=SD2(L3)+RZRL*(SD2(LJ)-SD2(L4))
      QUR=QU(L1)+RZRR*(QU(LJ)-QU(L2))
      QUL=QU(L3)+RZRL*(QU(LJ)-QU(L4))
*IF DEF,B64
      QUR=AMAX1(QUR,0.)
      QUL=AMAX1(QUL,0.)
*ENDIF
*IF DEF,B32
      QUR=DMAX1(QUR,0.D0)
      QUL=DMAX1(QUL,0.D0)
*ENDIF
      YYR=YY(L1)+RZRR*(YY(LJ)-YY(L2))
      YYL=YY(L3)+RZRL*(YY(LJ)-YY(L4))
      ZZR=ZZ(L1)+RZRR*(ZZ(LJ)-ZZ(L2))
      ZZL=ZZ(L3)+RZRL*(ZZ(LJ)-ZZ(L4))
      VAMUR=VAMU(LJ)
      QR=Q(L1)+RZRR*(Q(LJ)-Q(L2))
      QL=Q(L3)+RZRL*(Q(LJ)-Q(L4))
*IF DEF,B64
      QR=AMAX1(QR,0.0)
      QL=AMAX1(QL,0.0)
*ENDIF
*IF DEF,B32
      QR=DMAX1(QR,0.D0)
      QL=DMAX1(QL,0.D0)
*ENDIF
      VR=V(LJ)
      VL=VR
      DVR=DV(L1)+RZRR*(DV(LJ)-DV(L2))
      DVL=DV(L3)+RZRL*(DV(LJ)-DV(L4))
      CSR=CS(L1)+RZRR*(CS(LJ)-CS(L2))
      CSL=CS(L3)+RZRL*(CS(LJ)-CS(L4))
      CSR=ABS(CSR)
      CSL=ABS(CSL)
      QOR=QO(L1)+RZRR*(QO(LJ)-QO(L2))
      QOL=QO(L3)+RZRL*(QO(LJ)-QO(L4))
*IF DEF,B64
      QOR=AMAX1(QOR,0.)
      QOL=AMAX1(QOL,0.)
*ENDIF
*IF DEF,B32
      QOR=DMAX1(QOR,0.D0)
      QOL=DMAX1(QOL,0.D0)
*ENDIF
      YOZR=YOZ(L1)+RZRR*(YOZ(LJ)-YOZ(L2))
      YOZL=YOZ(L3)+RZRL*(YOZ(LJ)-YOZ(L4))
      ER=E(L1)+RZRR*(E(LJ)-E(L2))
      EL=E(L3)+RZRL*(E(LJ)-E(L4))
      EL=ABS(EL)
      ER=ABS(ER)
      DE=E(LJ)*ZM(LJ)/(ER*ZMR+EL*ZML)
      ER=ER*DE
      EL=EL*DE
      EADDR=EADD(L1)+RZRR*(EADD(LJ)-EADD(L2))
      EADDL=EADD(L3)+RZRL*(EADD(LJ)-EADD(L4))
      EADDL=ABS(EADDL)
      EADDR=ABS(EADDR)
      DE=0.0
      IF (EADD(LJ).NE.0.0) DE=EADD(LJ)*ZM(LJ)/(EADDR*ZMR+EADDL*ZML)
      EADDR=EADDR*DE
      EADDL=EADDL*DE
      PR=P(L1)+RZRR*(P(LJ)-P(L2))
      PL=P(L3)+RZRL*(P(LJ)-P(L4))
      PSMAXR=PSMAX(L1)+RZRR*(PSMAX(LJ)-PSMAX(L2))
      PSMAXL=PSMAX(L3)+RZRL*(PSMAX(LJ)-PSMAX(L4))
      PSMINR=PSMIN(L1)+RZRR*(PSMIN(LJ)-PSMIN(L2))
      PSMINL=PSMIN(L3)+RZRL*(PSMIN(LJ)-PSMIN(L4))
      PXR=PX(LJ)
      PXL=0.5*(PX(LM1)+PXR)
      TEMPR=TEMP(L1)+RZRR*(TEMP(LJ)-TEMP(L2))
      TEMPL=TEMP(L3)+RZRL*(TEMP(LJ)-TEMP(L4))
      TSPLLR=TSPALL(LJ)
      TSPLLL=TSPLLR
      ZFMR=ZFM(L1)+RZRR*(ZFM(LJ)-ZFM(L2))
      ZFML=ZFM(L3)+RZRL*(ZFM(LJ)-ZFM(L4))
      EIR=EI(L1)+RZRR*(EI(LJ)-EI(L2))
      EIL=EI(L3)+RZRL*(EI(LJ)-EI(L4))
      EIL=ABS(EIL)
      EIR=ABS(EIR)
      DE=0.0
      IF (EI(LJ).NE.0.0) DE=EI(LJ)*ZM(LJ)/(EIR*ZMR+EIL*ZML)
      EIR=EIR*DE
      EIL=EIL*DE
      FR=F(L1)+RZRR*(F(LJ)-F(L2))
      FL=F(L3)+RZRL*(F(LJ)-F(L4))
      FOR=FO(L1)+RZRR*(FO(LJ)-FO(L2))
      FOL=FO(L3)+RZRL*(FO(LJ)-FO(L4))
      NEL=NELEM(M)
      DO 130 NN=1,NEL
      ZFR(NN)=ZF(NN,L1)+RZRR*(ZF(NN,LJ)-ZF(NN,L2))
      ZFL(NN)=ZF(NN,L3)+RZRL*(ZF(NN,LJ)-ZF(NN,L4))
  130 CONTINUE
      DO 140 LL=1,NSPEC
      SSR(LL)=SS(L1,LL)+RZRR*(SS(LJ,LL)-SS(L2,LL))
      SSL(LL)=SS(L3,LL)+RZRL*(SS(LJ,LL)-SS(L4,LL))
  140 CONTINUE
```

```
      IF (JDBG.EQ.1) WRITE (6,730) RZRL,RZRR,XL,XR,UL,UR,TSPLLL,TSPLLR
C
C      SHIFT ZONAL PROPERTIES TO CORRESPOND TO NEW MESH
C
      JJJJ=JFIN-LJ
      DO 170 JJJ=1,JJJJ
      JJ=JFIN-JJJ+1
      IF (JDBG.EQ.1) WRITE (6,650) JJ
      X(JJ+1)=X(JJ)
      U(JJ+1)=U(JJ)
      ZM(JJ+1)=ZM(JJ)
      S(JJ+1)=S(JJ)
      SD(JJ+1)=SD(JJ)
      SD2(JJ+1)=SD2(JJ)
      QU(JJ+1)=QU(JJ)
      YY(JJ+1)=YY(JJ)
      VAMU(JJ+1)=VAMU(JJ)
      ZZ(JJ+1)=ZZ(JJ)
      Q(JJ+1)=Q(JJ)
      V(JJ+1)=V(JJ)
      DV(JJ+1)=DV(JJ)
      CS(JJ+1)=CS(JJ)
      QO(JJ+1)=QO(JJ)
      YOZ(JJ+1)=YOZ(JJ)
      E(JJ+1)=E(JJ)
      EADD(JJ+1)=EADD(JJ)
      P(JJ+1)=P(JJ)
      PSMAX(JJ+1)=PSMAX(JJ)
      PSMIN(JJ+1)=PSMIN(JJ)
      PX(JJ+1)=PX(JJ)
      TEMP(JJ+1)=TEMP(JJ)
      TSPALL(JJ+1)=TSPALL(JJ)
      ZFM(JJ+1)=ZFM(JJ)
      EI(JJ+1)=EI(JJ)
      F(JJ+1)=F(JJ)
      FO(JJ+1)=FO(JJ)
      NEL=NELEM(M)
      DO 150 NN=1,NEL
      ZF(NN,JJ+1)=ZF(NN,JJ)
  150 CONTINUE
      DO 160 LL=1,NSPEC
      SS(JJ+1,LL)=SS(JJ,LL)
  160 CONTINUE
  170 CONTINUE
C
C      INPUT PROPERTIES OF NEW ZONES INTO THE NEW MESH
C
      X(LP1)=XR
      X(LJ)=XL
      IF (JDBG.EQ.1) WRITE (6,790) LJ,X(LJ),X(LP1)
```

```
U(LP1)=UR
U(LJ)=UL
ZM(LP1)=ZMR
ZM(LJ)=ZML
S(LP1)=SR
S(LJ)=SL
SD(LP1)=SDR
SD(LJ)=SDL
SD2(LP1)=SD2R
SD2(LJ)=SD2L
QU(LP1)=QUR
QU(LJ)=QUL
VAMU(LP1)=VAMUR
VAMU(LJ)=VAMUR
YY(LP1)=YYR
YY(LJ)=YYL
ZZ(LP1)=ZZR
ZZ(LJ)=ZZL
Q(LP1)=QR
Q(LJ)=QL
V(LP1)=VR
V(LJ)=VL
DV(LP1)=DVR
DV(LJ)=DVL
CS(LP1)=CSR
CS(LJ)=CSL
QO(LP1)=QOR
QO(LJ)=QOL
YOZ(LP1)=YOZR
YOZ(LJ)=YOZL
E(LP1)=ER
E(LJ)=EL
EADD(LP1)=EADDR
EADD(LJ)=EADDL
P(LP1)=PR
P(LJ)=PL
PSMAX(LP1)=PSMAXR
PSMAX(LJ)=PSMAXL
PSMIN(LP1)=PSMINR
PSMIN(LJ)=PSMINL
PX(LP1)=PXR
PX(LJ)=PXL
TEMP(LP1)=TEMPR
TEMP(LJ)=TEMPL
TSPALL(LP1)=TSPLLR
TSPALL(LJ)=TSPLLL
ZFM(LP1)=ZFMR
ZFM(LJ)=ZFML
EI(LP1)=EIR
EI(LJ)=EIL
```

```
      F(LP1)=FR
      F(LJ)=FL
      FO(LP1)=FOR
      FO(LJ)=FOL
      NEL=NELEM(M)
      DO 180 NN=1,NEL
      ZF(NN,LP1)=ZFR(NN)
      ZF(NN,LJ)=ZFL(NN)
  180 CONTINUE
      DO 190 LL=1,NSPEC
      SS(LP1,LL)=SSR(LL)
      SS(LJ,LL)=SSL(LL)
  190 CONTINUE
      DO 200 LM=1,NMTRLS
      IF (JDBG.EQ.1) WRITE (6,740) J,LJ,JBND(LM)
      IF (LJ.GT.JBND(LM)) GO TO 200
      JBND(LM)=JBND(LM)+1
      IF (JDBG.EQ.1) WRITE (6,740) J,LJ,JBND(LM)
  200 CONTINUE
      IF (NJEDIT.EQ.0) GO TO 220
      DO 210 LL=1,NJEDIT
      IF (LJ.GT.JEDIT(LL)) GO TO 210
      JEDIT(LL)=JEDIT(LL)+1
  210 CONTINUE
  220 CONTINUE
      IF (LM1.LE.JHAT) JHAT=JHAT+1
      IF (LM1.LE.JSMAX) JSMAX=JSMAX+1
      IF (LM1.LE.JSTAR) JSTAR=JSTAR+1
      IF (LM1.LE.JTS) JTS=JTS+1
      JFIN=JFIN+1
      IF (JFIN.GE.200) GO TO 600
      J=J+1
  230 CONTINUE
      J=J+1
      IF (J.GE.JFIN) GO TO 600
  240 CONTINUE
      J=J+1
      JJPRIN=J
      GO TO 10
C
C       REZONE TO LEFT OF GRADIENT   (DIVIDE ZONES) ...
C
  250 CONTINUE
      JF=MAX0(JF,JJPRIN)
      JJJ=J-1-JF
      LL=1
      IF (JDBG.EQ.1) WRITE (6,750) J,JF
      DO 400 LLL=1,JJJ
      LJ=J-LLL+1
      LM1=LJ-1
      LP1=LJ+1
      ZMT=2.0*ZM(LJ)
      IF (ZM(LM1).GE.ZMT) GO TO 400
      IF (ZM(LP1).GE.ZMT) GO TO 400
      IF (JDBG.EQ.1) WRITE (6,760) LJ
      IF (LJ.LT.JJPRIN) GO TO 400
      XMIN=1.E-8*CS(LJ)
      DELX=X(LJ)-X(LM1)
      IF (DELX.LE.XMIN) GO TO 400
      XL=(X(LJ)+X(LM1))/2.0
      XR=X(LJ)
C
C      CHECK FOR MATERIAL BOUNDRIES ...
C
      DO 260 LM=1,NMTRLS
      IF (JDBG.EQ.1) WRITE (6,770) LJ,JBND(LM)
      IF (LJ.EQ.JBND(LM)) GO TO 280
      IF ((LM1).EQ.JBND(LM)) GO TO 270
  260 CONTINUE
      IF (TSPALL(LM1).EQ.1.234) GO TO 270
      IF (TSPALL(LJ).EQ.1.234) GO TO 400
      IF (TSPALL(LJ).EQ.0.) GO TO 270
      IF (TSPALL(LJ).NE.TSPALL(LM1)) GO TO 270
      IF (TSPALL(LJ).NE.TSPALL(LP1)) GO TO 280
C
C     DIVIDE INTERNAL ZONE
C
      RZRL=(XL-X(LJ-2))/(XR-X(LJ-2))
      RZRR=(X(LP1)-XL)/(X(LP1)-X(LM1))
      L1=LP1
      L2=LP1
      L3=LM1
      L4=LM1
      GO TO 290
C
C       DIVIDE ZONE WITH MATERIAL BOUNDRY AT LEFT INTERFACE ...
C
  270 CONTINUE
      RZRR=(X(LP1)-XL)/(X(LP1)-X(LM1))
      RZRL=1.-RZRR
      L1=LP1
      L2=LP1
      L3=LJ
      L4=LP1
      GO TO 290
C
C        DIVIDE ZONE WITH MATERIAL BOUNDRY AT RIGHT INTERFACE ...
C
  280 CONTINUE
      RZRL=(XL-X(LJ-2))/(XR-X(LJ-2))
      RZRR=1.-RZRL
      L1=LJ
      L2=LM1
      L3=LM1
      L4=LM1
  290 CONTINUE
      ZMR=0.5*ZM(LJ)
      ZML=ZMR
      UR=U(LJ)
      UL=(ZMR*U(LM1)+ZML*U(LJ))/ZM(LJ)
C
      IF (JDBG.EQ.1) WRITE (6,720) DR,DL,V(LM1),V(LJ),V(LP1),ZMR,ZML,ZM
     1 (LJ)
      SR=S(L1)+RZRR*(S(LJ)-S(L2))
      SL=S(L3)+RZRL*(S(LJ)-S(L4))
      SDR=SD(L1)+RZRR*(SD(LJ)-SD(L2))
      SDL=SD(L3)+RZRL*(SD(LJ)-SD(L4))
      SD2R=SD2(L1)+RZRR*(SD2(LJ)-SD2(L2))
      SD2L=SD2(L3)+RZRL*(SD2(LJ)-SD2(L4))
      QUR=QU(L1)+RZRR*(QU(LJ)-QU(L2))
      QUL=QU(L3)+RZRL*(QU(LJ)-QU(L4))
      VAMUR=VAMU(LJ)
*IF DEF,B64
      QUR=AMAX1(QUR,0.)
      QUL=AMAX1(QUL,0.)
*ENDIF
*IF DEF,B32
      QUR=DMAX1(QUR,0.D0)
      QUL=DMAX1(QUL,0.D0)
*ENDIF
      YYR=YY(L1)+RZRR*(YY(LJ)-YY(L2))
      YYL=YY(L3)+RZRL*(YY(LJ)-YY(L4))
      ZZR=ZZ(L1)+RZRR*(ZZ(LJ)-ZZ(L2))
      ZZL=ZZ(L3)+RZRL*(ZZ(LJ)-ZZ(L4))
      QR=Q(L1)+RZRR*(Q(LJ)-Q(L2))
      QL=Q(L3)+RZRL*(Q(LJ)-Q(L4))
*IF DEF,B64
      QR=AMAX1(QR,0.)
      QL=AMAX1(QL,0.)
*ENDIF
*IF DEF,B32
      QR=DMAX1(Q  ,0.D0)
      QL=DMAX1(QL,0.D0)
*ENDIF
      VR=V(LJ)
      VL=VR
      DVR=DV(L1)+RZRR*(DV(LJ)-DV(L2))
      DVL=DV(L3)+RZRL*(DV(LJ)-DV(L4))
      CSR=CS(L1)+RZRR*(CS(LJ)-CS(L2))
      CSL=CS(L3)+RZRL*(CS(LJ)-CS(L4))
      CSR=ABS(CSR)
      CSL=ABS(CSL)
      QOR=QO(L1)+RZRR*(QO(LJ)-QO(L2))
      QOL=QO(L3)+RZRL*(QO(LJ)-QO(L4))
*IF DEF,B64
      QOR=AMAX1(QOR,0.)
      QOL=AMAX1(QOL,0.)
*ENDIF
*IF DEF,B32
      QOR=DMAX1(QOR,0.D0)
      QOL=DMAX1(QOL,0.D0)
*ENDIF
      YOZR=YOZ(L1)+RZRR*(YOZ(LJ)-YOZ(L2))
      YOZL=YOZ(L3)+RZRL*(YOZ(LJ)-YOZ(L4))
      ER=E(L1)+RZRR*(E(LJ)-E(L2))
      EL=E(L3)+RZRL*(E(LJ)-E(L4))
      EL=ABS(EL)
      ER=ABS(ER)
      DE=E(LJ)*ZM(LJ)/(EL*ZML+ER*ZMR)
      EL=EL*DE
      ER=ER*DE
      EADDR=EADD(L1)+RZRR*(EADD(LJ)-EADD(L2))
      EADDL=EADD(L3)+RZRL*(EADD(LJ)-EADD(L4))
      EADDL=ABS(EADDL)
      EADDR=ABS(EADDR)
      DE=0.0
      IF (EADD(LJ).NE.0.0) DE=EADD(LJ)*ZM(LJ)/(EADDR*ZMR+EADDL*ZML)
      EADDL=EADDL*DE
      EADDR=EADDR*DE
      PR=P(L1)+RZRR*(P(LJ)-P(L2))
      PL=P(L3)+RZRL*(P(LJ)-P(L4))
      PSMAXR=PSMAX(L1)+RZRR*(PSMAX(LJ)-PSMAX(L2))
      PSMAXL=PSMAX(L3)+RZRL*(PSMAX(LJ)-PSMAX(L4))
      PSMINR=PSMIN(L1)+RZRR*(PSMIN(LJ)-PSMIN(L2))
      PSMINL=PSMIN(L3)+RZRL*(PSMIN(LJ)-PSMIN(L4))
      PXR=PX(LJ)
      PXL=0.5*(PX(LM1)+PXR)
      TEMPR=TEMP(L1)+RZRR*(TEMP(LJ)-TEMP(L2))
      TEMPL=TEMP(L3)+RZRL*(TEMP(LJ)-TEMP(L4))
      TSPLLR=TSPALL(LJ)
      TSPLLL=TSPLLR
      ZFMR=ZFM(L1)+RZRR*(ZFM(LJ)-ZFM(L2))
      ZFML=ZFM(L3)+RZRL*(ZFM(LJ)-ZFM(L4))
      EIR=EI(L1)+RZRR*(EI(LJ)-EI(L2))
      EIL=EI(L3)+RZRL*(EI(LJ)-EI(L4))
      EIL=ABS(EIL)
      EIR=ABS(EIR)
      DE=0.0
      IF (EI(LJ).NE.0.0) DE=EI(LJ)*ZM(LJ)/(EIR*ZMR+EIL*ZML)
      EIL=EIL*DE
      EIR=EIR*DE
      FR=F(L1)+RZRR*(F(LJ)-F(L2))
      FL=F(L3)+RZRL*(F(LJ)-F(L4))
      FOR=FO(L1)+RZRR*(FO(LJ)-FO(L2))
      FOL=FO(L3)+RZRL*(FO(LJ)-FO(L4))
      NEL=NELEM(M)
      DO 300 NN=1,NEL
      ZFR(NN)=ZF(NN,L1)+RZRR*(ZF(NN,LJ)-ZF(NN,L2))
      ZFL(NN)=ZF(NN,L3)+RZRL*(ZF(NN,LJ)-ZF(NN,L4))
  300 CONTINUE
      DO 310 LL=1,NSPEC
      SSR(LL)=SS(L1,LL)+RZRR*(SS(LJ,LL)-SS(L2,LL))
      SSL(LL)=SS(L3,LL)+RZRL*(SS(LJ,LL)-SS(L4,LL))
  310 CONTINUE
      IF (JDBG.EQ.1) WRITE (6,730) RZRL,RZRR,XL,XR,UL,UR,TSPLLL,TSPLLR
C
C      SHIFT ZONAL PROPERTIES TO CORRESPOND TO NEW MESH
C
      JJJJ=JFIN-LJ
      DO 340 JJQ=1,JJJJ
      JJ=JFIN-JJQ+1
      IF (JDBG.EQ.1) WRITE (6,780) JJ
      X(JJ+1)=X(JJ)
      U(JJ+1)=U(JJ)
      ZM(JJ+1)=ZM(JJ)
      S(JJ+1)=S(JJ)
      SD(JJ+1)=SD(JJ)
      SD2(JJ+1)=SD2(JJ)
      QU(JJ+1)=QU(JJ)
      VAMU(JJ+1)=VAMU(JJ)
      YY(JJ+1)=YY(JJ)
      ZZ(JJ+1)=ZZ(JJ)
      Q(JJ+1)=Q(JJ)
      V(JJ+1)=V(JJ)
      DV(JJ+1)=DV(JJ)
      CS(JJ+1)=CS(JJ)
      QO(JJ+1)=QO(JJ)
      YOZ(JJ+1)=YOZ(JJ)
      E(JJ+1)=E(JJ)
      EADD(JJ+1)=EADD(JJ)
      P(JJ+1)=P(JJ)
      PSMAX(JJ+1)=PSMAX(JJ)
      PSMIN(JJ+1)=PSMIN(JJ)
      PX(JJ+1)=PX(JJ)
      TEMP(JJ+1)=TEMP(JJ)
      TSPALL(JJ+1)=TSPALL(JJ)
      ZFM(JJ+1)=ZFM(JJ)
      EI(JJ+1)=EI(JJ)
      F(JJ+1)=F(JJ)
      FO(JJ+1)=FO(JJ)
      NEL=NELEM(M)
      DO 320 NN=1,NEL
      ZF(NN,JJ+1)=ZF(NN,JJ)
  320 CONTINUE
      DO 330 LL=1,NSPEC
      SS(JJ+1,LL)=SS(JJ,LL)
  330 CONTINUE
  340 CONTINUE
C
C       INPUT PROPERTIES OF NEW ZONES INTO THE NEW MESH
C
      X(LP1)=XR
      X(LJ)=XL
      IF (JDBG.EQ.1) WRITE (6,790) LJ,X(LJ),X(LP1)
      U(LP1)=UR
      U(LJ)=UL
      ZM(LP1)=ZMR
      ZM(LJ)=ZML
      S(LP1)=SR
      S(LJ)=SL
      SD(LP1)=SDR
      SD(LJ)=SDL
      SD2(LP1)=SD2R
      SD2(LJ)=SD2L
      QU(LP1)=QUR
      QU(LJ)=QUL
      VAMU(LP1)=VAMUR
      VAMU(LJ)=VAMUR
      YY(LP1)=YYR
      YY(LJ)=YYL
      ZZ(LP1)=ZZR
      ZZ(LJ)=ZZL
      Q(LP1)=QR
      Q(LJ)=QL
      V(LP1)=VR
      V(LJ)=VL
      DV(LP1)=DVR
      DV(LJ)=DVL
      CS(LP1)=CSR
      CS(LJ)=CSL
      QO(LP1)=QOR
      QO(LJ)=QOL
      YOZ(LP1)=YOZR
      YOZ(LJ)=YOZL
      E(LP1)=ER
      E(LJ)=EL
      EADD(LP1)=EADDR
      EADD(LJ)=EADDL
      P(LP1)=PR
      P(LJ)=PL
      PSMAX(LP1)=PSMAXR
      PSMAX(LJ)=PSMAXL
      PSMIN(LP1)=PSMINR
      PSMIN(LJ)=PSMINL
      PX(LP1)=PXR
      PX(LJ)=PXL
      TEMP(LP1)=TEMPR
      TEMP(LJ)=TEMPL
      TSPALL(LP1)=TSPLLR
      TSPALL(LJ)=TSPLLL
      ZFM(LP1)=ZFMR
      ZFM(LJ)=ZFML
      EI(LP1)=EIR
      EI(LJ)=EIL
      F(LP1)=FR
      F(LJ)=FL
      FO(LP1)=FOR
      FO(LJ)=FOL
      NEL=NELEM(M)
      DO 350 NN=1,NEL
      ZF(NN,LP1)=ZFR(NN)
      ZF(NN,LJ)=ZFL(NN)
  350 CONTINUE
      DO 360 LL=1,NSPEC
      SS(LP1,LL)=SSR(LL)
      SS(LJ,LL)=SSL(LL)
  360 CONTINUE
      DO 370 LM=1,NMTRLS
      IF (JDBG.EQ.1) WRITE (6,740) J,LJ,JBND(LM)
      IF (LJ.GT.JBND(LM)) GO TO 370
      JBND(LM)=JBND(LM)+1
      IF (JDBG.EQ.1) WRITE (6,740) J,LJ,JBND(LM)
  370 CONTINUE
      IF (NJEDIT.EQ.0) GO TO 390
      DO 380 LL=1,NJEDIT
      IF (LJ.GT.JEDIT(LL)) GO TO 380
      JEDIT(LL)=JEDIT(LL)+1
  380 CONTINUE
  390 CONTINUE
      IF (LJ.LE.JHAT) JHAT=JHAT+1
      IF (LJ.LE.JSMAX) JSMAX=JSMAX+1
      IF (LJ.LE.JSTAR) JSTAR=JSTAR+1
      IF (LJ.LE.JTS) JTS=JTS+1
      JFIN=JFIN+1
      IF (JFIN.GE.200) GO TO 600
      LL=LLL
      IF (JDBG.EQ.1) WRITE (6,800) JFIN
  400 CONTINUE
      J=J+LL
      JJPRIN=J
      GO TO 10
  410 CONTINUE
C
C     COMBINE ZONE IF ZONE J+1 IS NOT TOO HEAVY
C
      IF (ZM(J+1).GE.2.0*ZM(J)) GO TO 590
      IF (J+1.GE.JFIN) GO TO 420
      IF (ZM(J+1).GE.2.0*ZM(J+2)) GO TO 590
C
C     COMBINE ZONE IF ZONE J IS NOT TOO HEAVY
C
  420 IF (J.LE.2) GO TO 430
      IF (ZM(J).GE.2.0*ZM(J-1)) GO TO 590
  430 CONTINUE
      IF (ZM(J).GE.2.0*ZM(J+1)) GO TO 590
C
C      COMBINE ZONE  (IF NOT ON A MATERIAL BOUNDRY)
C
      DO 440 LM=1,NMTRLS
      IF (J.EQ.JBND(LM)) GO TO 590
  440 CONTINUE
C
C     COMBINE ZONE IF NOT AT A JEDIT LOCATION
C
      IF (NJEDIT.EQ.0) GO TO 460
      DO 450 LM=1,NJEDIT
      IF (J.EQ.JEDIT(LM)) GO TO 590
  450 CONTINUE
  460 CONTINUE
C
C     COMBINE ZONE IF NOT ON A PHASE OR SPALL BOUNDARY
C
      IF (TSPALL(J).NE.TSPALL(J+1)) GO TO 590
      IF (TSPALL(J).EQ.1.234) GO TO 590
      IF (TSPALL(J).EQ.0.) GO TO 590
      IF (TSPALL(J+1).EQ.1.234) GO TO 590
      IF (TSPALL(J+1).EQ.0.) GO TO 590
      IF (JFIN.LE.100) GO TO 590
      IF (JDBG.EQ.1) WRITE (6,810)
      X(J)=X(J+1)
      H1=(ZM(J-1)+ZM(J))*U(J-1)+(ZM(J)+ZM(J+1))*U(J)+(ZM(J+1)+ZM(J+2))*U
     1 (J+1)
      Z1=ZM(J-1)+ZM(J)+ZM(J+1)
      Z2=ZM(J)+ZM(J+1)+ZM(J+2)
      A1=(ZM(J+2)+Z1)*Z1
      A2=ZM(J-1)*U(J-2)*Z2-ZM(J+2)*U(J+2)*Z1-H1*(ZM(J-1)+Z1)
      E1=ZM(J-1)*(U(J-2)+U(J-1))*U(J-1)+ZM(J)*((U(J-1)+U(J))*U(J-1)+U(J)
     1 *U(J))+ZM(J+1)*((U(J)+U(J+1))*U(J)+U(J+1)*U(J+1))+ZM(J+2)*(U(J+1)
     2 +U(J+2))*U(J+1)
      A3=H1*(H1+ZM(J+2)*U(J+2))-Z2*E1
      UL=-.5*A2/A1
      UR=(H1-Z1*UL)/Z2
      DET=A2*A2-4.*A1*A3
      IF (DET) 470,480,480
  470 CONTINUE
C
C     UL AND UR HAVE COMPLEX VALUES.  WE MERELY TAKE THE REAL PART OF
C     UL AND UR AND CALCULATE THE RESULTING NONCONSERVATION OF KINETIC
C     ENERGY.  THIS NONCONSERVATION TERM WILL BE ADDED TO THE INTERNAL
C     ENERGY OF THE NEWLY COMBINED ZONE TO ENSURE COMPLETE CONSERVATION
C     OF ENERGY.
C
      DE=(E1-(ZM(J-1)*U(J-2)+Z1*UL)*UL-(ZM(J)+ZM(J+1))*UL*UR-(ZM(J+2)*U
     1 (J+2)+Z2*UR)*UR)/6.
      GO TO 500
  480 CONTINUE
C
C     UL AND UR ARE REAL.  HOWEVER BOTH UR AND UL CAN EACH TAKE ON ONE
C     OF TWO VALUES.  BOTH PAIRS OF VALUES SATISFY BOTH CONSERVATION OF
C     MOMENTUM AND CONSERVATION OF KINETIC ENERGY.  THEREFORE WE SELECT
C     THE PAIR OF VALUES WHICH IS CLOSEST TO THE VALUES OF U(J-1) AND
C     U(J+1).
C
      U1=.5*SQRT(DET)/A1
      U2=Z1*U1/Z2
      TEST=UL*U1+U2*U(J+1)-U1*U(J-1)-UR*U2
      IF (TEST.GT.0.) GO TO 490
      UL=UL+U1
      UR=UR-U2
      DE=0.
      GO TO 500
  490 CONTINUE
      UL=UL-U1
      UR=UR+U2
      DE=0.
  500 CONTINUE
      ZMINV=1.0/(ZM(J)+ZM(J+1))
      E(J)=(E(J)*ZM(J)+E(J+1)*ZM(J+1)+DE)*ZMINV
      S(J)=(S(J)*ZM(J)+S(J+1)*ZM(J+1))*ZMINV
      SD(J)=(SD(J)*ZM(J)+SD(J+1)*ZM(J+1))*ZMINV
      SD2(J)=(SD2(J)*ZM(J)+SD2(J+1)*ZM(J+1))*ZMINV
      QU(J)=(QU(J)*ZM(J)+QU(J+1)*ZM(J+1))*ZMINV
      VAMU(J)=(VAMU(J)*ZM(J)+VAMU(J+1)*ZM(J+1))*ZMINV
      YY(J)=(YY(J)*ZM(J)+YY(J+1)*ZM(J+1))*ZMINV
      ZZ(J)=(ZZ(J)*ZM(J)+ZZ(J+1)*ZM(J+1))*ZMINV
      Q(J)=(Q(J)*ZM(J)+Q(J+1)*ZM(J+1))*ZMINV
      V(J)=(V(J)*ZM(J)+V(J+1)*ZM(J+1))*ZMINV
      DV(J)=(DV(J)*ZM(J)+DV(J+1)*ZM(J+1))*ZMINV
      CS(J)=(CS(J)*ZM(J)+CS(J+1)*ZM(J+1))*ZMINV
      QO(J)=(QO(J)*ZM(J)+QO(J+1)*ZM(J+1))*ZMINV
      YOZ(J)=(YOZ(J)*ZM(J)+YOZ(J+1)*ZM(J+1))*ZMINV
      EADD(J)=(EADD(J)*ZM(J)+EADD(J+1)*ZM(J+1))*ZMINV
      P(J)=(P(J)*ZM(J)+P(J+1)*ZM(J+1))*ZMINV
      PX(J)=PX(J+1)
      PSMAX(J)=(PSMAX(J)*ZM(J)+PSMAX(J+1)*ZM(J+1))*ZMINV
      PSMIN(J)=(PSMIN(J)*ZM(J)+PSMIN(J+1)*ZM(J+1))*ZMINV
      TEMP(J)=(TEMP(J)*ZM(J)+TEMP(J+1)*ZM(J+1))*ZMINV
      ZFM(J)=(ZFM(J)*ZM(J)+ZFM(J+1)*ZM(J+1))*ZMINV
      EI(J)=(EI(J)*ZM(J)+EI(J+1)*ZM(J+1))*ZMINV
      F(J)=(F(J)*ZM(J)+F(J+1)*ZM(J+1))*ZMINV
      NNN=NELEM(M)
      DO 510 NN=1,NNN
      ZF(NN,J)=(ZF(NN,J)*ZM(J)+ZF(NN,J+1)*ZM(J+1))*ZMINV
  510 CONTINUE
      DO 520 LL=1,NSPEC
      SS(J,LL)=(SS(J,LL)*ZM(J)+SS(J+1,LL)*ZM(J+1))*ZMINV
  520 CONTINUE
      ZM(J)=ZM(J)+ZM(J+1)
      U(J-1)=UL
      U(J)=UR
C
C       SHIFT ZONES TO THE LEFT AFTER COMBINING ...
C
      JJJ=J+1
      DO 550 JJ=JJJ,JFIN
      X(JJ)=X(JJ+1)
      U(JJ)=U(JJ+1)
      ZM(JJ)=ZM(JJ+1)
      S(JJ)=S(JJ+1)
      SD(JJ)=SD(JJ+1)
      SD2(JJ)=SD2(JJ+1)
      QU(JJ)=QU(JJ+1)
      VAMU(JJ)=VAMU(JJ+1)
      YY(JJ)=YY(JJ+1)
      ZZ(JJ)=ZZ(JJ+1)
      Q(JJ)=Q(JJ+1)
      V(JJ)=V(JJ+1)
      DV(JJ)=DV(JJ+1)
      CS(JJ)=CS(JJ+1)
      QO(JJ)=QC(JJ+1)
      YOZ(JJ)=YOZ(JJ+1)
      E(JJ)=E(JJ+1)
      EADD(JJ)=EADD(JJ+1)
      P(JJ)=P(JJ+1)
      PX(JJ)=PX(JJ+1)
      PSMAX(JJ)=PSMAX(JJ+1)
      PSMIN(JJ)=PSMIN(JJ+1)
      TEMP(JJ)=TEMP(JJ+1)
      TSPALL(JJ)=TSPALL(JJ+1)
      ZFM(JJ)=ZFM(JJ+1)
      EI(JJ)=EI(JJ+1)
      F(JJ)=F(JJ+1)
      NNN=NELEM(M)
      DO 530 NN=1,NNN
      ZF(NN,JJ)=ZF(NN,JJ+1)
  530 CONTINUE
      DO 540 LL=1,NSPEC
      SS(JJ,LL)=SS(JJ+1,LL)
  540 CONTINUE
  550 CONTINUE
      DO 560 LM=1,NMTRLS
      IF (JDBG.EQ.1) WRITE (6,740) J,J,JBND(LM)
      IF (J.GT.JBND(LM)) GO TO 560
      IF (JBND(LM).NE.0) JBND(LM)=JBND(LM)-1
      IF (JDBG.EQ.1) WRITE (6,740) J,J,JBND(LM)
  560 CONTINUE
      IF (NJEDIT.EQ.0) GO TO 580
      DO 570 LM=1,NJEDIT
      IF (J.GT.JEDIT(LM)) GO TO 570
      IF (JEDIT(LM).NE.0) JEDIT(LM)=JEDIT(LM)-1
  570 CONTINUE
  580 CONTINUE
      IF (J.LE.JHAT) JHAT=JHAT-1
      IF (J.LE.JSMAX) JSMAX=JSMAX-1
      IF (J.LE.JSTAR) JSTAR=JSTAR-1
      IF (J.LE.JTS) JTS=JTS-1
      JFIN=JFIN-1
  590 CONTINUE
      J=J+1
      JJPRIN=J
C
C     RETURN TO INITIAL LOOP THRU MESH TO CHECK FOR COMBINE AND DIVIDE
C
      GO TO 10
  600 CONTINUE
      IF (JDBG.NE.1) GO TO 620
      DO 610 NJ=1,JFIN
      WRITE (6,820) NJ,X(NJ)
  610 CONTINUE
  620 CONTINUE
      J1=MAX0(JHAT,JSTAR)
      MS=1
      SK2M1=1.1*SK2M
      DO 640 J=2,J1
      DX=X(J)-X(J-1)
      IF (TSPALL(J).NE.1.234) GO TO 630
      MS=MS+1
      DX=XS(MS)-X(J-1)
  630 CONTINUE
*IF DEF,B64
```

```
      SK2M1=AMAX1(CS(J)/DX,SK2M1)
*ENDIF
*IF DEF,B32
      SK2M1=DMAX1(CS(J)/DX,SK2M1)
*ENDIF
  640 CONTINUE
*IF DEF,B64
      SK2M=AMAX1(SK2M,SK2M1)
*ENDIF
*IF DEF,B32
      SK2M=DMAX1(SK2M,SK2M1)
*ENDIF
      IF (JDBG.EQ.1) WTAPE=1.0
      CALL EDIT
      RETURN
C
  650 FORMAT (40H SHIFT ZONAL PROPS AFTER DIVIDE... JJ = ,I6)
  660 FORMAT (13H **990 JFIN =,I7)
  670 FORMAT (14H ***REZONE J =,I6)
  680 FORMAT (44H AAA, ARZCO, SLIM, DELS, ADELS, S(J), SMAX =,1P7E12.3/)
  690 FORMAT (44H *** DISTRZ, DTNH, CS(J), NREZON, U(J), AU =,/5X,1P3E11
     1 .3,I7,1P2E11.3/)
  700 FORMAT (22HO  J, LJ, JFIN, DELX =,3I7,1PE12.4)
  710 FORMAT (31HO  REZONE TO RIGHT  J, JF, LJ =,3I7)
  720 FORMAT (40H DR,DL,V(J-1),V(J),V(J+1),ZMR,ZML,ZM(J) /5X,1P8E12.4/)
  730 FQRMAT (43H RZRL, RZRR, XL, XR, UL, UR, TSPLLL, TSPLLR/5X,1P8E12.4
     1 /)
  740 FORMAT (21HO   J, LJ, JBND(LM) =,3I7)
  750 FORMAT (30H  REZONE LEFT OF GRADIENT FROM,I5,3H TO,I5/)
  760 FORMAT (12H    ***   LJ =,I8)
  770 FORMAT (18H REZON LEFT  LJ = ,I7,17H      JBND(LM) = ,I7)
  780 FORMAT (39H SHIFT ZONE PROP. AFTER DIVIDE..  JJ = ,I6)
  790 FORMAT (15H COMBINE   J = ,I6,7H X(J) =,1PE12.4,9H X(J+1) =,1PE12.
     1 4/)
  800 FORMAT (12H***** JFIN =,I6)
  810 FORMAT (18H COMBINE TWO ZONES/)
  820 FORMAT (4H J =,I7,6H   X =,E10.3)
      END
```

```
*DECK SAHA
      SUBROUTINE SAHA
*IF DEF,B32
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
*ENDIF
C
      REAL LGDEL
C
*CALL BLANK
*CALL AA
*CALL AC
*CALL EQFL
*CALL INDX
      DIMENSION A(7,10), RMAX(10)
C
C     THIS ROUTINE CALCULATES THE DEGREE OF IONIZATION IN THE MESHES
C             AND THE ENERGY FLOW ACROSS MESH BOUNDARIES WHEN GIVEN
C             MESH VOLUMES AND TEMPERATURES.
C
      DATA A, RMAX /70*0.,10*0./
*IF DEF,B64
      ZSTAR=AMAX1(ZFM(J),1.E-20)
*ENDIF
*IF DEF,B32
      ZSTAR=DMAX1(ZFM(J),1.D-20)
*ENDIF
      XNATOM=6.02E23/XMW(M)/V(J)
      FXPC=2.43E15*TEMP(J)**1.5/XNATOM
      IF (FXPC.GT.1.0) GO TO 20
C
C     HERE WE MAKE A ROUGH GUESS AT THE VALUE OF THE IONIZATION
C          WHEN IT IS SMALL
C
      ZSTAR=0.0
      DO 10 N1=1,NEM
      N=IELEM(M,N1)
C     AVOID EXP UNDERFLOW
      ARGEXP=-XI(N,1)/8.64E-5/TEMP(J)
      EXPR=0.0
      IF (ARGEXP.GT.-675.) EXPR=EXP(ARGEXP)
   10 ZSTAR=ZSTAR+SQRT(FXPC*EXPR)*AF(M,N)*FLOAT(NATOM(M))
C
C     FOR LARGER IONIZATIONS THE GUESSED VALUE IS TAKEN AS THE LAST
C         VALUE  --  ZFM(J)
C
   20 MCOUNT=0
      NPLUS=0
      DO 30 N1=1,NEM
      N=IELEM(M,N1)
   30 IPLUSO(N)=0
   40 NCHNG=1
      ZSTAR1=1.0
C
C     FIND 8 LEVELS - K - OF IONIZATION - NI(K,N) - FOR EACH ELEMENT
C      - N - OCCURRING IN MATERIAL M AND FIND THE RATIOS OF THE POPULA
C     TIONS - A(K,N) - IN THESE LEVELS
C
      DO 80 N1=1,NEM
      N=IELEM(M,N1)
C
C     CHANGE IN KMAX TO INCLUDE GROUND STATE AS ONE OF THE POSSIBLE
C     IONIZATION STATES
C
      KMAX(N)=MINO(NTBL(N)+1,8)
      KN=KMAX(N)
      IPLUS(N)=0
*IF DEF,B64
      IZF=IFIX(ZF(N,J))-KN/2
*ENDIF
*IF DEF,B32
      IZF=IDINT(ZF(N,J))-KN/2
*ENDIF
      IZF=MAXO(IZF,-1)
      IZF=MINO(IZF,NTBL(N)-KN)
      DO 50 K=1,KN
   50 NI(K,N)=IZF+K
      IF (J.LE.JTS.ANI.KPRIN.EQ.1) WRITE (6,220) N,(NI(K,N),K=1,8)
      KGO=KN-1
      KGO=MAXO(KGO,1)
      DO 70 K=1,KGO
      L=NI(K,N)+1
      AX1=XI(N,L)/(8.64E-5*TEMP(J))
      IF (AX1.LT.200.) GO TO 60
      A(K,N)=0.
      GO TO 70
   60 A(K,N)=FXPC*EXP(-AX1)
   70 IF (KPRIN.EQ.1.AND.J.EQ.JTS) WRITE (6,230) MATL(M),NAMEL(N),NI(K,N
     1 ),KGO,KN,L,J,XI(N,L),TEMP(J),A(K,N),FXPC
   80 CONTINUE
C
C     FIND FRACTION OF ATOMS - R(K,N) - IN EACH LEVEL - K - AND THE
C      LARGEST FRACTION IN EACH ELEMENT
C
   90 DO 150 N1=1,NEM
      N=IELEM(M,N1)
      KN=KMAX(N)
      KGO=KN-1
      IF (KN.LE.2) GO TO 100
      GO TO 110
```

```
  100 CONTINUE
      R(1,N)=0.
      A(1,N)=A(1,N)/ZSTAR*ZSTAR1
      IF (A(1,N).GT.0.  R(1,N)=0.5*A(1,N)*(SQRT(1.+4./A(1,N))-1.)
      IF (A(1,N).GT.1..AND.R(1,N).EQ.0.) R(1,N)=1.
      R(2,N)=R(1,N)
      RMAX(N)=R(1,N)
      IF (KPRIN.EQ.1.AND.J.EQ.JTS) WRITE (6,240) NAMEL(N),ZSTAR,ZSTAR1
     1 ,KN,KGO,A(1,N)
      GO TO 150
  110 CONTINUE
      DO 120 K=1,KGO
      A(K,N)=A(K,N)/ZSTAR*ZSTAR1
      IF (KPRIN.EQ.1.AND.J.EQ.JTS) WRITE (6,240) NAMEL(N),ZSTAR,ZSTAR1
     1 ,KN,KGO,A(K,N)
  120 CONTINUE
      R(1,N)=A(KGO,N)
      DO 130 K=2,KGO
      KGO1=KN-K
      R(1,N)=A(KGO1,N)*(1.0+R(1,N))
      IF (KPRIN.EQ.1.AND.J.EQ.JTS) WRITE (6,250) NAMEL(N),KGO1,K,A(KGO1
     1 ,N),R(1,N)
  130 CONTINUE
      R(1,N)=1./(1.+R(1,N))
      RMAX(N)=R(1,N)
      IF (KPRIN.EQ.1.AND.J.EQ.JTS) WRITE (6,260) NAMEL(N),R(1,N),RMAX(N)
      DO 140 K=2,KN
      R(K,N)=A(K-1,N)*R(K-1,N)
      IF (KPRIN.EQ.1) WRITE (6,270) NAMEL(N),K,A(K-1,N),R(K-1,N),R(K,N)
  140 IF (R(K,N).GT.RMAX(N)) RMAX(N)=R(K,N)
  150 RMAX(N)=0.1*RMAX(N)
C
C     SET OLD IONIZATION EQUAL TO ZSTAR1 AND CALCULATE A NEW VALUE
C     FOR EACH ELEMENT - ZF1(N) - AND A NEW TOTAL VALUE
C
      ZSTAR1=ZSTAR
      ZSTAR=0.0
      DO 170 N1=1,NEM
      N=IELEM(M,N1)
      KN=KMAX(N)
      ZF1(N)=0.0
      DO 160 K=1,KN
  160 ZF1(N)=ZF1(N)+R(K,N)*FLOAT(NI(K,N))
  170 ZSTAR=ZSTAR+AF(M,N)*ZF1(N)*FLOAT(NATOM(M))
C
C     IF NEW AND OLD VALUES OF TOTAL IONIZATION DIFFER BY LESS THAN
C     1 PERCENT NEW VALUE IS OK - ELSE A VALUE OF 0.5 * (ZNEW + ZOLD)
C       IS USED AND THE CALCULATION REPEATED
C
      MCOUNT=MCOUNT+1
      IF (MCOUNT.GT.100) GO TO 180
      IF (MCOUNT.GT.500) GO TO 210
      IF (ABS(ZSTAR-ZSTAR1).LT.0.01*ZSTAR1) GO TO 180
      IF (ZSTAR.LT.1.E-40) GO TO 210
      ZSTAR=(ZSTAR+ZSTAR1)/2.0
      GO TO 90
C
C     CHECK TO SEE IF LARGEST POSSIBLE EIGHT IONIZATION LEVELS HAVE BEEN
C     CHOSEN FOR EACH ELEMENT - IF NOT POOREST CHOSEN ELEMENT NUMBER
C      IS TAGGED  NC
C
  180 XMAX1=0.0
      DO 190 N1=1,NEM
      N=IELEM(M,N1)
      KN=KMAX(N)
      IF (KN.EQ.1) GO TO 190
      IF (FLOAT(NI(1,N))*R(1,N).GT.FLOAT(NI(KN-1,N))*R(KN-1,N).AND.NI(1
     1 ,N).GT.0) IPLUS(N)=-1
      IF (FLOAT(NI(KN,N))*R(KN,N).GT.FLOAT(NI(2,N))*R(2,N).AND.NI(KN,N)
     1 .LT.NTBL(N)) IPLUS(N)=1
      IF (IPLUS(N).EQ.0.OR.IPLUS(N)+IPLUSO(N).EQ.0) GO TO 190
      NCHNG=2
      XMAX=ABS(ZF1(N)-ZF(N,J))
      IF (XMAX.LT.XMAX1) GO TO 190
      XMAX1=XMAX
      NC=N
  190 CONTINUE
C
C     IONIZATION LEVELS IN ELEMENT NC ARE UPGRADED OR DOWNGRADED BY 1
C     AND THE ENTIRE CALCULATION OF ZFM(J) IS REDONE
C
      GO TO (210,200), NCHNG
      CALL GOTOER
  200 ZF(NC,J)=ZF(NC,J)+FLOAT(IPLUS(NC))
      NPLUS=NPLUS+1
      IPLUSO(NC)=IPLUS(NC)
      GO TO 40
  210 RETURN
C
  220 FORMAT ('ONI',I5/(1X,10I5))
  230 FORMAT ('0 SAHA',2(X,A10),' NI(K,N) =',I3,' KGO =',I3,' KN =',I3,'
     1 L =',I3,' J ='.I3,' XI(N,L) =',1PE10.3,' TEMP(J) =',1PE10.3,/10X.
     2 ' A(K,N) =',1PE10.3,' FXPC =',1PE10.3/)
  240 FORMAT ('0 SAHA ',A10,' ZSTAR =',1PE10.3,' ZSTAR1 =',1PE10.3,' KN
     1=',I3,' KGO =',I3,' A(K,N) =',1PE10.3/)
  250 FORMAT ('0 SAHA ',A10,' KGO1 =',I3,' K =',I3,' A(KGO1,N) =',1PE10.
     1 3,' R(1,N) =',1PE10.3/)
  260 FORMAT ('0 SAHA ',A10,' R(1,N), RMAX(N) =',1P2E10.3/)
  270 FORMAT ('0 SAHA ',A10,' K, A(K-1,N), R(K-1,N), R(K,N) =',I3,1P3E10
     1 .3/)
```

```
      END
```

```
*DECK SPALL
      SUBROUTINE SPALL
*IF DEF,B32
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
*ENDIF
C
*CALL BLANK
*CALL SPLLC
      E1=0.
      LL=1
      M=1
      MS=1
      SMAX=0.
      TMAX=0.
      DO 340 J=2,JFIN
C
C     CHANGE MATERIAL INDEX AND ADD NEW ACTIVE ZONE
C
      IF (J-JBND(M)) 20,10,20
   10 CONTINUE
      LL=LL+1
   20 CONTINUE
      IF (J.LT.JFIN) GO TO 30
      GO TO 340
C
C     CHECK FOR PREVIOUSLY SPALLED ZONE
C
   30 CONTINUE
      IF (TSPALL(J).EQ.1.234) GO TO 40
      GO TO 70
   40 CONTINUE
      ISPALL=3
C
C     NOW CHECK TO SEE IF ZONE SHOULD BE RECOMBINED
C
      IF (XS(MS).LT.X(J)) GO TO 70
C
C     YES, RECOMBINE ZONE
C
      U(J)=(U(J)*ZM(J+1)+US(MS)*ZM(J))/(ZM(J)+ZM(J+1))
      X(J)=X(J)+(XS(MS)-X(J))*ZM(J)/(ZM(J)+ZM(J+1))
      DU=U(J)-U(J-1)
      V(J)=(X(J)-X(J-1))/ZM(J)
      ISM=ISM-1
      IF (MS-1.EQ.ISM) GO TO 60
      DO 50 I=MS,ISM
      IP1=I+1
      XS(I)=XS(IP1)
      US(I)=US(IP1)
   50 CONTINUE
   60 CONTINUE
      TSPALL(J)=0.0
      ISPALL=1
      ISMP1=ISM+1
      XS(ISMP1)=0.0
      US(ISMP1)=0.0
      WRITE (6,410) J,X(J),N,TIME
      LINE=LINE+2
      IF (LINE.LE.50) GO TO 70
      WRITE (6,440)
      LINE=0
   70 CONTINUE
      IF (V(J).LT.0.0) WRITE (6,430) J,MS,N,XS(MS),X(J),X(J-1),TSPALL(J)
C
C     NOW CHECK FOR TENSION IN VAPOR AND MELT
C
      IF (TSPALL(J).EQ.8.) GO TO 290
      IF (TSPALL(J).EQ.7.) GO TO 80
      IF (TSPALL(J-1).NE.7..AND.TSPALL(J-1).NE.8.) GO TO 100
      IF (TSPALL(J).EQ.1.234.OR.TSPALL(J).EQ.0.0) GO TO 90
      GO TO 100
   80 CONTINUE
      IF (TSPALL(J+1).EQ.7.) GO TO 90
      IF (TSPALL(J+1).EQ.1.234.OR.TSPALL(J+1).EQ.0.0) GO TO 90
      GO TO 100
   90 CONTINUE
*IF DEF,B64
      S(J)=AMAX1(S(J),0.)
*ENDIF
*IF DEF,B32
      S(J)=DMAX1(S(J),0.DO)
*ENDIF
      GO TO 290
C
C     CHECK FOR FRACTURE IN SOLID MATERIAL
C
  100 CONTINUE
      IF (S(J)+Q(J).GE.0..OR.S(J-1)+Q(J-1).GE.0.) GO TO 290
      IF (TSPALL(J-1).EQ.8.) GO TO 290
      IF (TSPALL(J).EQ.1.234.OR.TSPALL(J).EQ.0.) GO TO 290
      IF (ISPALL.NE.0.OR.TSPALL(J-1).EQ.1.234) GO TO 290
      GO TO (110,120,130,140,150,160,170,180,190,200), ISPLLM(M)
      CALL GOTOER
  110 CONTINUE
C
C     FRACTURE MODEL BASED ON TENSION.  CALCULATE STRESS FOR COMPARISON
C     WITH TSPALL.
C
      SJ=((S(J)+Q(J))*ZM(J)+(S(J-1)+Q(J-1))*ZM(J-1))/(ZM(J)+ZM(J-1))
      IF (TSPALL(J-1)+SJ.GE.0.) GO TO 290
      IF (MS.LE.49) GO TO 210
      WRITE (6,450)
      STOP 21
  120 CONTINUE
C
C     SIMPLE STRAIN MODEL FOR FRACTURE.  SJ = AVERAGE STRAIN.
C
      E2=V(J)*RHO(M)
      E2=1./E2-1.
      SJ=.5*(E1+E2)
      E1=E2
      IF (TSPALL(J-1)+SJ.GE.0.) GO TO 290
      IF (MS.LE.49) GO TO 210
      WRITE (6,450)
      STOP 21
  130 CONTINUE
  140 CONTINUE
  150 CONTINUE
  160 CONTINUE
  170 CONTINUE
  180 CONTINUE
  190 CONTINUE
  200 CONTINUE
      GO TO 290
  210 CONTINUE
      ISM=ISM+1
      II=ISM
      JS=J-1
      XCAL=0.
      SUMASS=0.
      TSUMAS=0.
      M=1
      DO 250 I=2,JFIN
      IF (I-JS) 220,230,260
  220 CONTINUE
      IF (JBND(M).EQ.0) GO TO 240
      IF (I-JBND(M)) 240,230,230
  230 CONTINUE
      TSUMAS=TSUMAS+ZM(I)
      TCAL=TSUMAS/RHO(M)
      XCAL=XCAL+TCAL
      SUMASS=SUMASS+TSUMAS
      TSUMAS=0.
      M=M+1
      GO TO 250
  240 CONTINUE
      TSUMAS=TSUMAS+ZM(I)
  250 CONTINUE
  260 CONTINUE
```

```
      WRITE (6,400) JS,X(JS),N,TIME,SJ,TSPALL(JS),XCAL,SUMASS,ISM
      IF (TSPALL(JS).EQ.7..OR.TSPALL(JS-1).EQ.7.) S(JS)=0.
      TSPALL(JS)=1.234
      IF (TSPALL(JS-1).EQ.8..OR.TSPALL(J).NE.7.) IS=1
      LINE=LINE+3
      IF (LINE.LT.150) GO TO 280
      WRITE (6,440)
      LINE=0
      GO TO 280
  270 CONTINUE
      III=II-1
      XS(II)=XS(III)
      US(II)=US(III)
      II=III
  280 CONTINUE
      IF (II.GT.MS) GO TO 270
      XS(MS)=X(JS)
      US(MS)=U(JS)
      ISPALL=2
  290 CONTINUE
C
C     CALCULATE TMAX AND SMAX
C
      SJ=S(J)
      IF (SJ-TMAX) 300,310,310
  300 CONTINUE
      TMAX=SJ
      JTMAX=J
  310 CONTINUE
      IF (SJ-SMAX) 330,330,320
  320 CONTINUE
      SMAX=SJ
      JSMAX=J
  330 CONTINUE
      IF (ISPALL.GE.2) MS=MS+1
      ISPALL=0
      M=LL
  340 CONTINUE
      IF (IS.LE.0) RETURN
      IS=0
      I=0
      MS=1
      SMM=0
      DO 380 J=2,JFIN
      IF (TSPALL(J).NE.1.234) GO TO 350
      SMM=SMM+.5*ZM(J)*(U(J-1)+US(MS))
      MS=MS+1
      GO TO 360
  350 CONTINUE
      SMM=SMM+.5*ZM(J)*(U(J-1)+U(J))
  360 CONTINUE
      IF (J.EQ.JFIN) GO TO 370
      IF (TSPALL(J).EQ.8..AND.TSPALL(J+1).NE.8.) GO TO 370
      IF (TSPALL(J).NE.1.234) GO TO 380
  370 CONTINUE
      I=I+1
      SM(I)=SMM
      SMM=0.
  380 CONTINUE
      WRITE (6,420) (SM(J),J=1,I)
      I=I/8
      LINE=LINE+I+3
      IF (LINE.LE.50) GO TO 390
      WRITE (6,440)
      LINE=0
  390 CONTINUE
      RETURN
C
  400 FORMAT (/27H  === SPALL OCCURED AT ZONE,I4,9H LOCATION,1PE12.4,12H
     1 CM AT CYCLE,I4,6H TIME=,1PE11.4,11H SEC   SJ =,1PE11.4,11H TSPALL
     2(J)=,1PE11.4/5X,20HLOCATION IN MATERIAL,21H COORDINATE SYSTEM IS,1
     3 PE12.4,23H CM  TOTAL MASS TO THIS,8H POINT =,1PE11.4,26H GM  TOTA
     4L NO OF SPALLS IS,I3)
  410 FORMAT (/19H  $$$ COMBINED ZONE,I4,9H LOCATION,1PE12.4,13H CM  AT
     1CYCLE,I4,6H TIME=,1PE11.4,4H SEC)
  420 FORMAT (/24H MOMENTUM AFTER FRACTURE/(1P8E14.4))
  430 FORMAT (23H  NEGATIVE DENSITY ZONE,I4,5H  MS=,I3,10H  CYCLE NO,I4/
     1 9H  XS(MS)=,1PE14.4,7H  X(J)=,1PE14.4,9H  X(J-1)=,1PE14.4,12H TSP
     2ALL(J) =,1PE14.4)
  440 FORMAT (1H1)
  450 FORMAT (/51H   PROGRAM HAS REACHED THE NO OF SPALLS DIMENSIONED)
      END
```

```
*DECK TRANSP
      SUBROUTINE TRANSP
*IF DEF,B32
      IMPLICIT DOUBLEPRECISION(A-H,O-Z)
*ENDIF
*CALL BLANK
*CALL AA
*CALL AC
*CALL EQFL
*CALL INDX
      M=1
      LL=1
      NEM=NELEM(1)
      J1=MAX0(JSTAR,JHAT)
      J1=MIN0(J1,JFIN)
      DO 50 J=2,J1
      IF (J.EQ.JBND(M)) LL=M+1
      IF (ITER(J).EQ.0) GO TO 40
      XLAM1(J)=0.0
      XLAM2(J)=0.0
      IF (ZFM(J).LT.0.01) GO TO 40
      IF (ITER(J).EQ.1) GO TO 20
      Z1=0.0
      Z2=0.0
      DO 10 N1=1,NEM
      N=IELEM(M,N1)
      Z1=Z1+AF(M,N)*ZF(N,J)
   10 Z2=Z2+AF(M,N)*ZF(N,J)**2
      XLAM1(J)=3.18E-6*Z1/Z2/LGDEL(M)/TEMP(J)**0.5
      IF (ITER(J).EQ.2) GO TO 40
   20 IF (ZFM(J).LT.1.0) GO TO 40
      XLAMA=0.
      XLAMP=1.
C
C     NOW WE CALCULATE THE ROSELAND MEAN OPACITY TO BE STORED IN XLAM2(J
C
      DO 30 N1=1,NEM
      N=IELEM(M,N1)
*IF DEF,B64
      Z1=AMAX1(ZF(N,J),1.E-30)
*ENDIF
*IF DEF,B32
      Z1=DMAX1(ZF(N,J),1.D-30)
*ENDIF
      Z2=Z1*Z1+.25
      XLAMT=5.8E-26*(XMW(M)*V(J)/FLOAT(NATOM(M))/AF(M,N))**2*TEMP(J)**3.
     1 5/Z1/Z2
      XLAMP=XLAMP*XLAMT
   30 XLAMA=XLAMA+XLAMT
```

```
      IF (NEM.EQ.1) XLAMA=1.0
      XLAMP=XLAMP/XLAMA*7.56E-5
      XLAM2(J)=XLAMP
   40 M=LL
      NEM=NELEM(M)
   50 CONTINUE
      RETURN
C
      END
```

HYPUF SOURCE LISTING

HYPUF CORRECTION DECK

```
*ID RUN
*DEFINE B32
```

APPENDIX B
HYPUF INPUT INSTRUCTIONS

The attached table gives the input instructions for HYPUF in the form of a logic flow diagram. With the exception of those variables which require character input, all input is in free format. The variables that require character input are so designated in the instruction table (e.g., by : A10).

The user should be aware that one disadvantage of free format input is that something must be entered for every variable requested. Simply leaving the fields blank will not work. If an input should be zero, than enter 0. The input for a sample problem is provided following the input instructions table.

Following the sample problem input listing is a set of three graphs which summarize the calculated results for the sample problem. the first graph shows the time evolution of two measures of specific impulse which are calculated by the code. The specific impulse history is printed out on tape 9 (also known as FORTRAN Unit 9). A spreadsheet program was used to select the columns from the printout that were plotted. The second graph shows the calculated stress history at the midplane of the material. The last graph show the peak stress calculated as a function of position. Compressive stresses are positive in HYPUF. The sample problem given should be adequate for verifying the installation of HYPUF on a given computer system.

HYPUF INPUT INSTRUCTIONS

```
DISCPT(I), I=1,8 (FORMAT= 8A10)
     TITLE CARD IN CHARACTER FORMAT
---------------------------------------------------------------------------------------
NSPEC, NTEDT, NJEDIT, LOZHIZ, ILOG, ILIN, ICON, IDIF
     NSPEC = NUMBER OF RADIATION SOURCE SPECTRA
     NTEDT = NUMBER OF TIMES AT WHICH MESH EDITS ARE DESIRED.
               IF NTEDT IS NON-ZERO, PROVIDE NTEDT VALUES OF TEDIT.
     NJEDIT = NUMBER OF LOCATIONS AT WHICH STRESS HISTORY IS TO BE DETERMINED.
               IF NJEDIT IS NON-ZERO, PROVIDE NJEDIT PAIRS OF VALUES OF MTLN AND DSTF.
     LOZHIZ = FLAG TO INDICATE OF PROBLEM CONSISTS OF A LOW Z MATERIAL OVER A HIGH Z MATERIAL.
               IF SO SET LOZHIZ EQUAL TO 1, OTHERWISE SET LOZHIZ EQUAL TO Ø.
     ILOG = FLAG TO INDICATE THAT LOGARITHMIC PLOTS ARE DESIRED WHEN SET EQUAL TO 1.
               IF NO PLOTS ARE DESIRED, SET ILOG EQUAL TO 1.
     ILIN = FLAG TO INDICATE THAT LINE PRINTER PLOTS OF THE STRESS PROFILE
               ARE DESIRED WHEN SET EQUAL TO 1; OTHERWISE SET ILIN EQUAL TO Ø.
     ICON = FLAG TO INDICATE THAT THERMAL CONDUCTIVITY IN SOLID MATERIALS IS
               DESIRED WHEN SET EQUAL TO Ø; OTHERWISE SET ICON EQUAL TO 1.
     IDIF = FLAG TO INDICATE THAT THERMAL DIFFUSION IS TO BE INCLUDED WHEN
               SET EQUAL TO Ø;  OTHERWISE SET IDIF EQUAL TO 1.
---------------------------------------------------------------------------------------
                                  IF NTEDT .GT. Ø
     THEN                                                                   ELSE
-------------------------------------------|-------------------------------------------
(TEDIT(I), I=1,NTEDT)                      |
     TEDIT = PROBLEM TIME AT WHICH MESH EDIT|
               IS DESIRED.  PROVIDE NTEDT   |
               VALUES.  SKIP IF NTEDT = Ø.  |
---------------------------------------------------------------------------------------
                                  IF NJEDIT .GT. Ø
     THEN                                                                   ELSE
-------------------------------------------|-------------------------------------------
(MTLN(I), DSTF(I), I=1,NJEDIT)             |
     MTLN = MATERIAL LAYER NUMBER IN WHICH THE
               STRESS HISTORY PLOT IS DESIRED.
     DSTF = DISTANCE FRACTION (FROM LEFT   |
               BOUNDARY) OF MATERIAL LAYER AT
               WHICH STRESS HISTORY PLOT IS |
               DESIRED.                     |
     PROVIDE NJEDIT PAIRS OF MTLN AND DSTF. |
NRZC, NMTRLS, JRZL, JZPUL, NPRIN, NTAPE, NREZON, JCYCS
      NRZC = ZONING CONTROL.  IF HAND ZONING IS DESIRED, SET NRZC
             EQUAL TO THE NUMBER OF ZONE REGIONS AND PROVIDE JBND(M)
             THROUGH RZ(M) WHERE REQUIRED.
      NMTRLS = NUMBER OF MATERIAL LAYERS IN THE PROBLEM.
             PROVIDE A COMPLETE MATERIAL DESCRIPTION FOR EACH LAYER.
      JRZL = NO LONGER USED DIRECTLY.  IF RZCØ=Ø ON THE NEXT CARD
             JRZL WILL DETERMINE THE DEFAULT VALUE OF RZCØ.
             DEFAULT VALUE OF JRZL IS 25.
      JZPUL = USED TO DETERMINE THE DEFAULT VALUE OF RSCRIT ON THE NEXT CARD.
      NPRIN = EDIT IS CALLED EVERY NPRIN HYDRO CYCLES.  IF ILOG=Ø
             OR ILIN=Ø, EDIT WILL PRODUCE A LINE PRINTER PLOT ON TAPE 3.
             DEFAULT VALUE IS NPRIN=2Ø.
      NTAPE = SAME MEANING AS NPRIN. DEFAULT VALUE IS NTAPE=1ØØ.
      NREZON = SUBROUTINE REZONE IS CALLED EVERY NREZON CYCLES.
             DEFAULT VALUE IS NREZON=JCYCS.  THE DEFAULT VALUE
             WILL INHIBIT REZONING.
      JCYCS = MAXIMUM NUMBER OF HYDRO CYCLES ALLOWED.  RECOMMENDED
             VALUE IS JCYCS=5ØØØ.
-------------------------------------------------------------------------------
RSCRIT, RZCØ, RZC1
     RSCRIT = CRITICAL VALUE OF ZONE TO ZONE VARIATION IN STRESS
             IF RSCRIT IS TOO SMALL REZONE WILL TRY TO DIVIDE THE MESH TOO FINELY.
             IF RSCRIT IS TOO LARGE REZONE WILL COMBINE ZONES TO MAKE A COARSE
             MESH.  RECOMMENDED VALUE IS FROM .Ø2 TO .1.  DEFAULT VALUE IS 1/FLOAT(JZPUL).
     RZCØ = CRITICAL FRACTION OF PEAK STRESS.  RSCRIT DETERMINES ALLOWABLE
             ZONE TO ZONE VARIATIONS AWAY FROM THE MAIN PULSE.  RZCØ DETERMINES
             ALLOWABLE VARIATIONS IN THE NEGHBORHOOD OF THE MAIN PULSE.  RECOMMENDED
             VALUE IS .Ø5 TO .1.  DEFAULT VALUE IS 1/FLOAT(JRZL).
     RZC1 = REZONE NORMALIZES ALLOWABLE ZONE TO ZONE VARIATION IN STRESS TO THE
            ·DIFFERENCE BETWEEN THE PEAK STRESS AND RZC1.  RECOMMENDED VALUE IS Ø.
             DEFAULT VALUE IS Ø.
-------------------------------------------------------------------------------
CKS, TS, ANGLE, DTMIN, DIFTST, NDEP, JPRIN, ION
     CKS = STRESS DISTANCE CHECK.  WHEN THE PEAK STRESS HAS PROPAGATED TO
             X=CKS THE PROBLEM IS STOPPED.  RECOMMENDED VALUE IS SOME POINT BEYOND
             THE REAR SURFACE OF THE PROBLEM UNLESS THE USER WANTS TO STOP THE
             CALCULATION WHEN THE PEAK STRESS REACHES A GIVEN DISTANCE PART WAY
             THROUGH THE PROBLEM.
     TS = STOP TIME.  WHEN THE PROBLEM TIME EQUALS OR EXCEEDS TS, THE
             HYDROCALCULATION IS STOPPED AND ANY REQUESTED PLOTS ARE MADE.
     ANGLE = THE ANGLE FROM THE NORMAL AT WHICH THE INCIDENT RADIATION
             STRIKES THE MATERIAL.
     DTMIN = DETERMINES THE ACCURACY TO WHICH THE TEMPERATURES MUST BE
             CALCULATED IN SUBROUTINE EQST.  RECOMMENDED VALUE IS .Ø2.
     DIFST = NOT CURRENTLY USED.  INPUT Ø.
     NDEP = ENERGY DEPOSITION IN THE ZONES IS RECALCULATED EVERY NDEP HYDROCYCLES.
     JPRIN = CYCLE NUMBER AT WHICH DEBUGGING OUTPUT IS TO BEGIN.  INPUT JPRIN=0
             TO INHIBIT DEBUGGING.  OTHERWISE THE DEBUG PRINTOUTS WILL BEGIN AT
             HYDROCACLULATION CYCLE JPRIN.
     ION = DETERMINES WHETHER ELECTRON REARRANGEMENT IS ALLOWED DURING IONIZATION.
             INPUT ION=Ø TO PROHIBIT ELECTRON REARRANGEMENT.  INPUT ION=1 TO PERMIT
             REARRANGEMENT OF INNER SHELL ELECTRONS AS THE OUTER ELECTRONS ARE
             REMOVED BY IONIZATION.
```

HYPUF INPUT INSTRUCTIONS (CONTINUED)

```
IF NRZC .GT. Ø
THEN
    IF NMTRLS .GT. 1
    THEN
        (JBND(M), M=1,NMTRLS-1)
              JBND(M) = ZONE
                 NUMBERS OF THE
                 BOUNDARY OF ALL
                 BUT THE LAST
                 MATERIAL LAYER.
    ELSE
    JFIN, (NZ(L), L=1,NRZC)
          JFIN = NUMBER OF THE LAST ZONE IN THE PROBLEM
          NZ(L) = ZONE NUMBER OF THE LAST ZONE IN EACH REGION.
    DX, TIME, (RZ(M), M=1,NRZC)
          DX = SIZE OF THE FIRST ZONE IN THE PROBLEM
          TIME = INITIAL TIMESTEP.
          RZ(M) = ZONE TO ZONE SIZE RATIO IN EACH REGION
ELSE
    TIME, (RZ(M), M=1,NMTRLS)
          TIME = INITIAL TIME STEP. RECOMMENDED VALUE
             IS 1.E-12
          RZ(M) = THICKNESS IN CM OF EACH MATERIAL
             LAYER IN THE PROBLEM.  LATER REDEFINED
             TO BE THE ZONE TO ZONE SIZE RATIO BY
             THE PROGRAM.
NELT
      NELT = NUMBER OF ELEMENTS IN THE PROBLEM.  MUST BE 1Ø OR LESS.
(NOE(N), N=1,NELT)
      NOE(N) = NUMBER OF X-RAY ABSORBTION EDGES FOR EACH ELEMENT
FOR N = 1, NELT
    NAMEL (N)  :A1Ø
          NAMEL = ELEMENT NAME
    NTBL(N), NVARE(N), XAW(N)
          NTBL(N) = ATOMIC NUMBER OF THE ELEMENT
          NVARE(N) = VARIABLE X-RAY CROSS SECTION FLAG.  IF NVARE=Ø, VARIABLE
             CROSS-SECTION ARE PERMITTED IF NVARM=Ø FOR THE MATERIAL ALSO.
             IF NVARE=1, VARIABLE CROSS SECTIONS ARE PROHIBITED FOR THIS ELEMENT.
          XAW(N) = ATOMIC WEIGHT OF THIS ELEMENT.
    (AA(N,I), B(N,I), EDGE(N,I), I=1, NOE(N))
          X-RAY ABSORBTION COEFFICIENTS.  X-RAY ABSORBTION DUE TO A GIVEN SHELL
          IS AA*(PHOTON ENERGY)**B IF THE PHOTON ENERGY IS GREATER THAN THE EDGE
          ENERGY, EDGE.
    IF NVARE .GT. Ø
    THEN
        (XI(N,J), J=1,NTBL(N))
              XI(N,J) = X-RAY IONIZATION POTENTIALS FOR
                 ELEMENT N.
    ELSE
```

HYPUF INPUT INSTRUCTIONS (CONTINUED)

```
FOR M   MATL(M) : A10
= 1,         MATL(M) = MATERIAL NAME OF THE M'TH MATERIAL.
NMTRLS
        NVARM(M), NATOM(M)
             NVARM(M) = VARIABLE X-RAY CROSS-SECTION FLAG FOR MATERIAL M.
                IF NVARM(M)=0, VARIABLE CROSS SECTIONS ARE ALLOWED FOR EVERY ELEMENT
                FOR WHICH NVARE(N)=0.  IF NVARM(M)=1, VARIABLE CROSS-SECTIONS ARE
                PROHIBITED FOR THIS MATERIAL.
             NATOM(M) = THE NUMBER OF ATOMS PER MOLECULE.  USED TO ESTIMATE THE
                SPECIFIC HEAT OF THE MATERIAL BY THE DULONG-PETIT LAW.

        RHO(M), EQSTC(M), EQSTD(M), EQSTE(M), EQSTG(M), EQSTH(M), EQSTS(M), PMIN(M), ISPLLM(M), EM(M)
             RHO(M) = MATERIAL DENSITY (GM/CM-CU)
             EQSTC(M) CONSTANTS USED IN THE EQUATION OF STATE ROUTINE EQST.  FOR THE SOLID STATE
             EQSTD(M) THE HUGONIOT PRESSURE IS PH=(((EQSTS(M)*EMU+EQSTD(M))*EMU+EQSTC(M))*EMU)
             EQSTE(M)     *(1-.5*EQSTG(M)*EMU/ENU) WHERE ENU=1/(V*RHO(M)) AND EMU=ENU-1.
             EQSTG(M) FOR THE VAPOR PHASE EQSTH(M) IS THE THE RATIO OF SPECIFIC HEATS MINUS 1
             EQSTH(M) AND EQSTE(M) IS THE SUBLIMATION ENERGY.
             EQSTS(M)
             PMIN(M) = DYNAMIC SPALL STRENGTH OF THE MATERIAL.  INPUT AT A NEGATIVE PRESSURE
                (DYNES/CM-SQ)
             ISPLLM(M) = SPALL MODEL NUMBER. MODEL NUMBER 1 IS THE SPALL MODEL BASED ON
                SIMPLE TENSION AS USED IN PUFF74.  UP TO NINE OTHER MODELS CAN
                BE INSERTED IN THE CODE, BUT ONLY MODEL 1 HAS BEEN USED TO DATE.
             EM(M) = MELT ENERGY OF THE MATERIAL.  USED IN SPALL ROUTINE TO DETERMINE
                WHEN DYNAMIC SPALL STRENGHT GOES TO ZERO.

        CUSP1(M), CUSPA(M), CUSPC(M), CUSPD(M), CUSPG(M), CUSPS(M)
             PARAMETERS FOR A CUSPED EQUATION OF STATE.  CUSP1(M) IS THE PRESSURE AT
             WHICH AN INFLECTION OCCURS IN THE HUGONIOT DATA.  CUSPA(M) IS THE VALUE
             OF EMU AT WHICH THE INFLECTION OCCURS.  BEYOND THE INFLECTION POINT,
             THE HUGONIOT PRESSURE IS PH=(CUSP1(M)+CUSPC(M)*(EMU-CUSPA(M))+(CUSPD(M)
                *(EMU-CUSPA(M))**2+CUSPS(M)*(EMU-CUSPA)**3)*(1.-.5*(CUSPG(M)*EMU)

        Y0(M), AMU(M), YADD(M), YMU(M), XMW(M), LGDEL(M), XCON(M)
             Y0(M) = ROOM TEMPERATURE COMPRESSIVE YIELD STRENGTH.
             AMU(M) = THE SHEAR MODULUS
             YADD(M) = THE INCREASE IN YIELD STRENGHT DUE TO COMPRESSION BEYOND YMU.
             YMU(M) = THE VALUE OF EMU BEYOND WHICH PLASTIC FLOW OCCURS.
             XMW(M) = MOLECULAR WEIGHT OF THE MATERIAL.
             LGDEL = PARAMETER USED IN SUBROUTINE FLOION TO CALCULATE ELECTRON
                THERMAL CONDUCTIVITY.  RATHER THAN CALCULATING LGDEL, FLOION USES
                THE INPUT VALUE AS AN EFFECTIVE VALUE FOR THE MATERIAL.  RECOMMENDED
                VALUES ARE 5. TO 10.
             XCONN(M) = ROOM TEMPERATURE THERMAL CONDUCTIVITY IN ERGS/CM/DEG-K/S.
                EXAMPLES ARE:
                       ALUMINUM:  2.37E7
                       COPPER:  4.01E7
                       GOLD:  3.17E7
                       TITANIUM:  2.19E6
                       IRON:  8.02E6
                       CORK:  5.02E3
                       RUBBER:  1.3E4
```

HYPUF INPUT INSTRUCTIONS (CONTINUED)

```
|         |-----------------------------------------------------------------------------------------|
|         | EQSTA(M), OMEGA(M), PRELAX(M), SHEAR(M), TRELAX(M), MFLAG(M)                            |
|         |      EQSTA(M) = NONDIMENSIONAL CONSTANT USED IN BADE GEOMETRIC DISPERSION MODEL.        |
|         |      OMEGA(M) = CHARACTERISTIC FRQUENCY (INVERSE SECONDS) USED IN BADE MODEL.           |
|         |      PRELAX = CHARACTERISTIC PRESSURE (DYNE/CM-SQ) USED IN ELASTIC-VISCOPLASTIC MODEL.  |
|         |      SHEARR(M) = SHAPE PARAMETER (NONDIMENSIONAL) USED IN ELASTIC-VISCOPLASTIC MODEL.   |
|         |      TRELAX(M) = CHARACTERISTIC RELAXATION TIME (S) USED IN ELASTIC-VISCOPLASTIC MODEL. |
|         |      MFLAG(M) = FLAG TO INDICATE IF MAXWELL DISPERSION MODEL IS TO BE USED              |
|         |         INSTEAD OF BADE GEOMETRIC DISPERSION MODEL.  IF MFLAG=Ø, USE THE BADE MODEL.    |
|         |         IF MFLAG=1, PROVIDE CONSTANTS FOR MAXWELL MODEL.                                |
|         |-----------------------------------------------------------------------------------------|
|         |                                 IF MFLAG .EQ. 1                                         |
|         | THEN                                                                          ELSE      |
|         |------------------------------------------------------------|----------------------------|
|         | AMU2(M), TRELX2(M)                                         |                            |
|         |      AMU2(M) = EFFECTIVE SHEAR MODULUS (DYNE/CM-SQ)        |                            |
|         |      TRELX2(M) = RELAXATION TIME (S)                       |                            |
|         |------------------------------------------------------------|                            |
|         | NELEM(M)                                                                                |
|         |      NUMBER OF ELEMENTS IN THE MATERIAL.  CANNOT BE MORE THAN 5.                        |
|         |-----------------------------------------------------------------------------------------|
|         | (IELEM(M,N), N=1,NELEM(M))                                                              |
|         |      IELEM(M,N) = ELEMENT NUMBERS IN THE MATERIAL.  REFERS TO THE 1ST 2ND OR            |
|         |         NTH ELEMENT IN THE ORDER THE ELEMENTS WERE DESCRIBED ABOVE.                     |
|         |         DOES NOT REFER TO ATOMIC NUMBER OF THE ELEMENTS.                                |
|         |-----------------------------------------------------------------------------------------|
|         | (AF(M,IELEM(M,N)), N=1,NELEM(M))                                                        |
|         |      AF(M,IELEM(M,N)) = ATOMIC FRACTIONS OF THE ELEMENTS SPECIFIED IN THE               |
|         |         PREVIOUS CARD.  DOES NOT REFER TO WEIGHT FRACTIONS AS USED IN PUFF74            |
|         |         OR OTHER CODES.  REFERS TO RELATIVE ATOMIC NUMBER FRACTIONS INSTEAD.            |
|-------------------------------------------------------------------------------------------------|
| FOR NS  | NHNU(NS), NBB(NS)                                                                       |
| = 1,    |      NHNU = NUMBER OF ENERGY BINS IN THE INPUT RADIATION SPECTRUM.  IF A                |
|  NSPEC  |         BLACKBODY SPECTRUM OR MIXTURE OF BLACKBODY SPECTRA IS BEING USED                |
|         |         SET NHNU(NS)=Ø, OTHERWISE SET NHNU(NS)=THE NUMBER OF ENERGY BINS                |
|         |         USED TO DESCRIBE THE SPECTRUM.                                                  |
|         |      NBB(NS) = NUMBER OF BLACKBODY SOURCES IN THE SPECTRUM.  SET NBB(NS)=1              |
|         |         IF AN ARBITRARY SPECTRUM IS BEING USED.  OTHERWISE NBB(NS) IS THE               |
|         |         NUMBER OF BLCKBODY SOURCES UP TO A MAXIMUM OF THREE.                            |
|         |-----------------------------------------------------------------------------------------|
|         | START(NS), SSTOP(NS), (T(KK), E(KK), KK=1, NBB(NS)), XPRIN                              |
|         |      START(NS) = START TIME FOR RADIATION SOURCE NS.  HYPUF CAN HANDLE UP TO THREE SOURCE SPECTRA.
|         |      SSTOP(NS) = STOP TIME FOR SOURCE NS.                                               |
|         |      T(KK) = BLACKBODY TEMPERATURE IN KEV OF THE SOURCE.  IF AN ARBITRARY               |
|         |         SPECTRUM IS BEING USED, SET T(KK)=1.                                            |
|         |      E(KK) = SOURCE STRENGTH IN CAL/CM-SQ.                                              |
|         |      XPRIN = FLAG TO INDICATE OF SOURCE SPECTRUM PRINTOUT IS DESIRED.                   |
|         |         IF XPRIN=Ø. THE SOURCE SPECTRUM WILL BE PRINTED OUT.  IF XPRIN=1.               |
|         |         THE SOURCE SPECTRUM WILL NOT BE PRINTED OUT.                                    |
|         |                                                                                         |
```

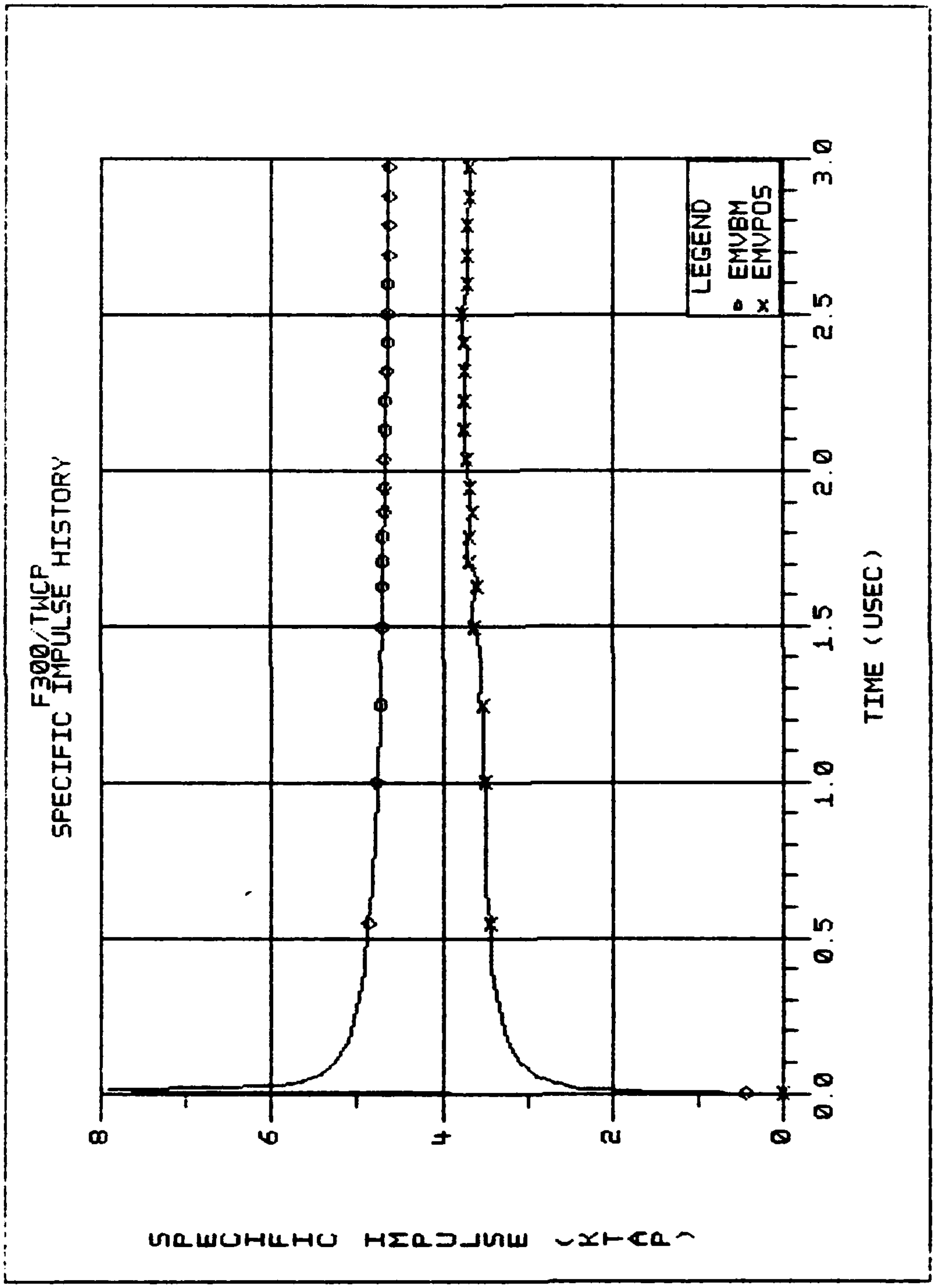
HYPUF INPUT INSTRUCTIONS (CONTINUED)

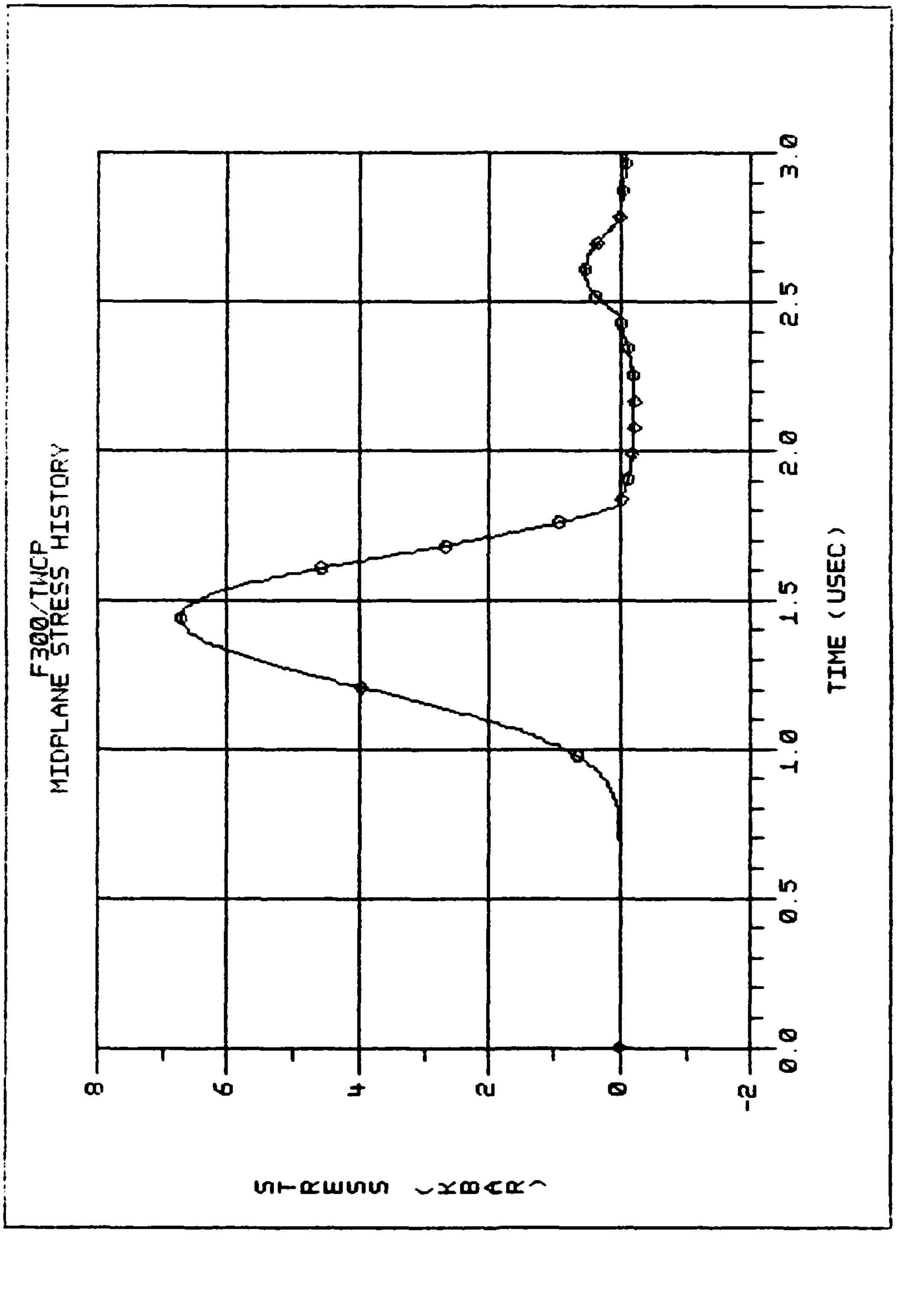
```
IF NHNU(NS) .EQ. Ø
THEN
    (TBL(I), ES(I), I=1, NHNU(NS))
        TBL(I) = CENTER OF THE I'TH ENERGY BIN IN THE
           SPECTRUM
        ES(I) = FRACTION OF TOTAL ENERGY SPECTRUM
           WHICH IS IN THE I'TH BIN.
ELSE
```

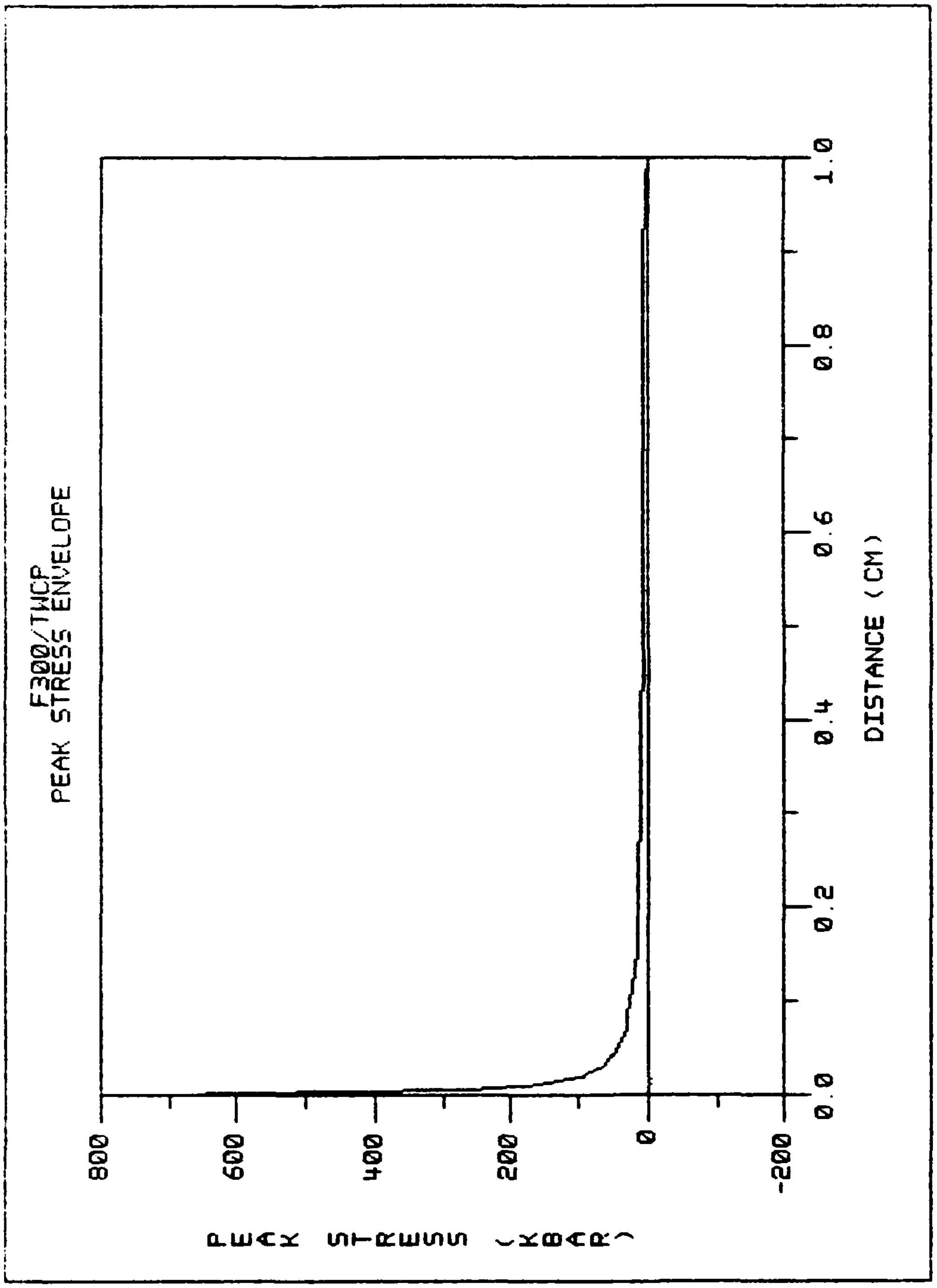
```
*** F300 / TCWP ***

1  5  1  1  1  1  0  0

1.E-10  5.E-10  1.E-9  2.E-9  5.1E-9

1  1.

-1  2  50  40  20  0  0  50000

.1   .125    0.

5.  3.0E-6  0.  .02  0.  10  0  1

1.E-12  .5  .5

4

2  4  4  4

HYDROGEN

1  0  1.0079

18.01  -3.006  .014

0.  0.  300.

CARBON

6  0  12.0111

21.77  -2.843  .005

39.93  -2.843  .013

2191.3   -2.436  .284

0.  0.  300.

NITROGEN

7  0  14.0067

72.76  -2.791  .007

45.64  -2.791  .018

2967.  -2.701  .4
```

```
0.  0.  300.
OXYGEN
8  0  15.9994
150.7  -2.707  .009
85.6  -2.707  .024
3619.   -2.768  .533
0.  0.  300.
TCWP
0  1
1.44  1.59E11  1.94E11  1.113E10  .52  .11  0.  -1.E9  1  205.
6*0.
3.16E9  3.1E10  0.  4.81E-2  12.085  5.  5.E3
6*0
4
1  2  3  4
.0229  .8915  .0077  .0779
TCWP
0  1
1.44  1.59E11  1.94E11  1.113E10  .52  .11  0.  -1.E9  1  205.
6*0.
3.16E9  3.1E10  0.  4.81E-2  12.085  5.  5.E3
6*0
4
1  2  3  4
.0229  .8915  .0077  .0779
2  1
```

```
0.  5.1E-9  1.  300.  0.

.49975  0.

1.  i.
```


F300/TWCP
SPECIFIC IMPULSE HISTORY
SPECIFIC IMPULSE (KTAP)
TIME (USEC)
0
2
4
6
8
0.0
0.5
1.0
1.5
2.0
2.5
3.0
LEGEND
◇ EMVBM
x EMVPOS


F300/TWCP
MIDPLANE STRESS HISTORY
STRESS (KBAR)
TIME (USEC)
8
6
4
2
0
-2
0.0
0.5
1.0
1.5
2.0
2.5
3.0


F300/TWCP
PEAK STRESS ENVELOFE
PEAK STRESS (KBAR)
800
600
400
200
0
-200
DISTANCE (CM)
0.0
0.2
0.4
0.6
0.8
1.0

DISTRIBUTION LIST

DNA-TR-89-25

DEPARTMENT OF DEFENSE

DEFENSE ADVANCED RSCH PROJ AGENCY
ATTN: F PATTEN

DEFENSE INTELLIGENCE AGENCY
ATTN: RTS-2B

DEFENSE NUCLEAR AGENCY
ATTN: SPSD
ATTN: SPSP/MAJ WOLF
ATTN: SPWE
4 CYS ATTN: TITL

DEFENSE TECHNICAL INFORMATION CENTER
2 CYS ATTN: DTIC/FDAB

STRATEGIC DEFENSE INITIATIVE ORGANIZATION
ATTN: T/KT LTC C MARTIN

DEPARTMENT OF THE ARMY

U S ARMY MATERIAL TECHNOLOGY LABORATORY
ATTN: R FITZPATRICK

U S ARMY STRATEGIC DEFENSE COMMAND
ATTN: SDC/E MONTGOMERY

DEPARTMENT OF THE NAVY

NAVAL RESEARCH LABQRATORY
ATTN: CODE 4600 D NAGEL
ATTN: CODE 4633 E FRIEBELE

OFFICE OF NAVAL TECHNOLOGY
ATTN: CODE 217

DEPARTMENT OF THE AIR FORCE

AIR FORCE CTR FOR STUDIES & ANALYSIS
ATTN: AFCSA/SAMI R GRIFFIN

FOREIGN TECHNOLOGY DIVISION, AFSC
ATTN: TQX-1, J TUSS

STRATEGIC AIR COMMAND/XRFS
ATTN: XRFS

WEAPONS LABORATORY
ATTN: NTC C AEBY
ATTN: TA LT COL W MULZER
ATTN: TA LT COL HERRERA
ATTN: TALE T EDWARDS
ATTN: TAS MAJ SQUILLER

WRIGHT RESEARCH & DEVELOPMENT CENTER
ATTN: MLPJ J RHODEHAMEL
ATTN: MLP W WOODY
ATTN: MLPJ R RONDEAU

DEPARTMENT OF ENERGY

LAWRENCE LIVERMORE NATIONAL LAB
ATTN: F SERDUKE
ATTN: L-84 H KRUGER
ATTN: M GERRISSIMENRO

LOS ALAMOS NATIONAL LABORATORY
ATTN: B259 A GREENE
ATTN: C931 T KING
ATTN: E548 R S DINGUS

SANDIA NATIONAL LABORATORIES
ATTN: DIV 8242 M BIRNBAUM
ATTN: DIV 8242 M MCDONALD

SANDIA NATIONAL LABORATORIES
ATTN: K MATZEN
ATTN: ORG 1230 J E POWELL

DEPARTMENT OF DEFENSE CONTRACTORS

AEROSPACE CORP
ATTN: H BLAES M2-250
ATTN: R COOPER
ATTN: T PARK

APTEK, INC
ATTN: DR E FITZGERALD

BATTELLE MEMORIAL INSTITUTE
ATTN: C WALTERS

GENERAL ELECTRIC CO
ATTN: D ENLOW

GENERAL RESEARCH CORP
ATTN: J SOMERS

JAYCOR
ATTN: M TREADAWAY

KAMAN SCIENCES CORP
ATTN: N FERRITER
ATTN: R ALMASSY

KAMAN SCIENCES CORPORATION
ATTN: DASIAC

KAMAN SCIENCES CORPORATION
ATTN: DASIAC

KTECH CORP
ATTN: D KELLER

MCDONNELL DOUGLAS CORPORATION
2 CYS ATTN: D JOHNSON
ATTN: J S KIRBY
ATTN: L THOMPSON

PACIFIC-SIERRA RESEARCH CORP
ATTN: H BRODE

PACIFICA TECHNOLOGY
ATTN: H JANEE

PHYSICAL SCIENCES, INC
ATTN: P WU

R & D ASSOCIATES
ATTN: B GOULD
ATTN: D GAKENHEIMER
ATTN: P A MILES

R & D ASSOCIATES
ATTN: J WALTON

S-CUBED
ATTN: G GURTMAN

SCIENCE APPLICATIONS INTL CORP
ATTN: E TOTON
ATTN: R AIREY
ATTN: S METH
ATTN: W CHADSEY

SPARTA, INC
ATTN: J E LOWDER
ATTN: R G ROOT

SRI INTERNATIONAL
ATTN: B HOLMES
ATTN: D CURRAN
ATTN: G ABRAHAMSON